

DIPLOMARBEIT

Titel der Diplomarbeit

„Ein innovatives Konzept zur Vermittlung von Informatik
in der AHS Oberstufe und dessen Umsetzung mit
Joomla!“

Verfasser

Florian Netsch

angestrebter akademischer Grad

Magister der Naturwissenschaften (Mag. rer. nat.)

Wien, 2014

Studienkennzahl lt. Studienblatt:

A 190 884 299

Studienrichtung lt. Studienblatt:

Lehramtsstudium UF Informatik und
Informatikmanagement
UF Psychologie und Philosophie

Betreuerin:

ao. Univ.-Prof. Dipl.-Ing. Dr. Renate Motschnig

INHALT

I.	Vorbemerkung.....	7
II.	Konzept für ein Schulbuch.....	13
1.	Vorwort	13
1.1.	Was Sie in diesem Buch nicht finden werden	13
1.2.	So wird der Informatik-Lehrplan abgedeckt	15
1.3.	Was Sie in diesem Buch finden werden	20
2.	Joomla! im Unterricht:	22
2.1.	Einfache Unterrichtseinheiten:	22
2.1.1.	Einrichtung von Joomla!:.....	22
2.1.2.	Inhaltliche Struktur eines CMS:	23
2.1.3.	Die Blackbox Joomla! wird zerlegt:	26
2.1.3.1.	Templates, Menüs und andere Erweiterungen oberflächlich betrachtet:	26
2.1.3.2.	Datenbanken kurz angeschaut... ..	27
2.1.4.	Das Zusammenspiel von HTML und CSS	28
2.1.5.	Weg von der „Klick-Informatik“: PHP	30
2.1.6.	Wie funktioniert das mit den DIV-Containern in HTML?	31
2.1.7.	Ein Formular mit PHP	34
2.1.7.1.	Zunächst das Formular ohne PHP	35
2.1.7.2.	PHP ist eine Hürde:.....	36
2.2.	Barrierefreie Webseiten im Unterricht:	43
2.2.1.	Zur Theorie:	43
2.2.2.	Ein praktisches Beispiel:	45
2.2.3.	Strukturierung als Unterrichtsprojekt:	47
2.2.4.	Barriereelos zur Handyansicht:	48
2.2.4.1.	Diskussion der Möglichkeiten:	49

2.3. Fazit:	50
2.4. Komplexe Arbeitsaufgaben mit und für Joomla!:	50
2.4.1. Die erste eigene Komponente:.....	51
2.4.2. Das Template:.....	55
2.4.3. Das Joomla-Framework.....	66
2.4.3.1. Was versteht man unter dem Framework?	66
2.4.3.2. Programmiere eine Komponente für eine Stellenanzeige:.....	68
2.4.3.3. Fazit:	68
2.5. Unterrichtspraktikum in Salzstiegl:	69
2.5.1. Vorbemerkung:.....	69
2.5.2. Arbeitsplan:	69
2.5.3. Fazit:	71
2.6. Jahresplanung:.....	71
2.6.1. Vorbemerkung:.....	71
2.6.2. Fünfte Klasse:	72
2.6.3. Sechste Klasse:	72
2.6.4. Siebente Klasse:.....	73
2.6.5. Achte Klasse:	73
2.6.6. Nachbemerkung:.....	73
2.7. Anforderungen für die LehrerInnen-Ausbildung.....	73
3. Schluss	77
3.1. Abstract (Zusammenfassung).....	77
3.2. Nachbemerkung	77
3.3. Was ist gelungen, was ist misslungen?	79
3.4. Code-Beispiele.....	80
3.4.1. HTML:	80
3.4.2. CSS:.....	81
3.4.3. PHP, SQL und Datenbanken:	82

3.4.4.Joomla!-Code:	84
3.5. Wozu braucht man GIMP?	84
3.6. Lebenslauf	87
3.7. Literaturverzeichnis.....	89

I. Vorbemerkung

Die vorliegende Arbeit soll als Konzept zur Entwicklung eines Schulbuches für den Informatik-Unterricht an der Oberstufe der AHS dienen. Ein ambitioniertes Vorhaben, wenn man bedenkt, dass es schon seit Jahren keine Schulbücher mehr für den Informatikunterricht gibt und das aus gutem Grund: Niemand will festlegen, was man eigentlich aus der immer größer werdenden Auswahl an Möglichkeiten wirklich an der Schule in Informatik unterrichten soll. Die Ziele und Inhalte des Informatikunterrichts unterliegen nicht nur im Laufe der Jahre einem logischen Wandel, der mit der Entwicklung neuer Technologien zusammenhängt, sie hängen nicht selten natürlich auch von persönlichen Neigungen und Kenntnissen des Lehrers ab.

Eine Zeit lang galten für einige LehrerInnen die Module des ECDL¹ als gute Arbeitsgrundlage für den Unterricht, andere meinten wiederum, dass diese einfache Aneignung von Skills wenig mit Informatik zu tun hätte, wo es doch darum ginge, dass man lernt Probleme mit den richtigen Algorithmen möglichst selbstständig zu lösen. Der Lehrplan für Informatik² schafft einen überraschend breiten Rahmen, der durch den ECDL keineswegs gefüllt werden kann.

Zur Zeit hat man allerdings das Gefühl, dass dem engagierten Projekt Informatik-Unterricht an der AHS etwas die Luft ausgeht. Alles begann ja im Schuljahr 1985/86 mit der Einführung des Faches Informatik als Verbindliche Übung, die 1989/90 zum Pflichtfach Informatik für alle SchülerInnen der 5. Klassen umgewandelt wurde.³ Das Fach wurde damals – und vielfach auch noch heute – von interessierten LehrerInnen unterrichtet, die teilweise eine recht kurze, teilweise eine etwas längere (max. ca. zwei Jahre, minimal wenige Wochenenden) Ausbildung dafür gemacht hatten. Viele LehrerInnen mussten auch wegen ihren mangelnden Kenntnissen SchülerInnen unterfordern, bzw. wurden sie von diesen rasch überfordert. Diverse Arbeitsgemeinschaften haben um 2009 ihr Engagement langsam verloren.⁴ Selbst bei

¹ Europäischer Computerführerschein. Vgl. <http://www.ecdl.at>

² Vgl. http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf

³ <http://www.gym1.at/schulinformatik/buecher/ahs-informatik.pdf> S. 2 und S. 5.

⁴ Vgl. <http://it1018.egger.ac/> oder <http://www.arge-informatik.tsn.at/> oder

sehr produktiv wirkenden Projekten hat man das Gefühl, dass zumindest mit 2012 der Faden gerissen ist.⁵

So ist der vom Ministerium abgesegnete vorläufige und noch nicht offizielle Leitfaden zur Matura Neu wieder sehr breit angelegt.⁶ Man hat das Gefühl, dass dort LehrerInnen jeweils versucht haben, ihre Unterrichtsmodelle auch einzubringen. Altbewährtes wurde in den Leitfaden hineingeschrieben, wenn auch schon teilweise überholt, wie die Idee, als Prüfungsaufgabe von SchülerInnen eine LAN-Party organisieren zu lassen, Computer zu Netzwerken zusammenzustecken oder vielleicht ein Legomännchen über den Tisch laufen zu lassen.

Woran liegt diese Orientierungslosigkeit vieler InformatiklehrerInnen heute?

Mit Sicherheit an der Rasanz der Entwicklung, die tatsächlich atemlos macht: Was vorgestern unterrichtet wurde, war gestern ungenau, ist heute falsch und morgen völlig absurd. Man denke z. B. nur an die Idee, Webseiten mit Tabellen zu erstellen, was vor 10 Jahren noch gängige Praxis war und heute nach den Gesetzen der Barrierefreiheit absoluter Unsinn ist.

Es ist aber auch so, dass sich der Bereich der Informatik sehr rasch erweitert und eine unglaubliche Spezialisierung der Fachleute nötig macht, die LehrerInnen aber total überfordern muss, weil sie ja eigentlich über alle wichtigen Bereiche Bescheid wissen sollten, tatsächlich aber meist nicht einmal in wesentlichen Bereichen einigermaßen professionell arbeiten können, da hierfür eine Ausbildung, die nur einige Wochenenden gedauert hat, niemals ausreichen kann.⁷ Ich denke, ein Informatiklehrer sollte auch daran gemessen werden, ob er mit seinen Kenntnissen in der Privatwirtschaft Geld verdienen könnte.

Vor diesem Horizont ist die Verzagttheit mancher KollegInnen nachvollziehbar. Die Problematik der AHS-LehrerInnen trifft meiner Meinung nach im gleichen Umfang die UniversitätslehrerInnen, die ja eigentlich für eine zeitgemäße Ausbildung der AHS-LehrerInnen sorgen sollten, selbst aber gar nicht immer die Zeit aufbringen können, sich

⁵ Vgl. z. B. <http://www.ahs-informatik.com/>

⁶ <https://www.box.com/s/mrkqjercceOrpxt93hdn>

⁷ Vgl. z. B.: Birgit Weyand, Monika Justus, Michael Schratz (Hrsg.): Auf unsere Lehrerinnen und Lehrer kommt es an. Geeignete Lehrer/-innen gewinnen, (aus-)bilden und fördern. Essen: 2012. S. 15.

an die neuesten Trends der Informatik anzupassen. So gibt es auch immer wieder die Diskussion zwischen den VerteterInnen, die davon überzeugt sind, die „richtige“ Informatik zu unterrichten und den VertreterInnen, die immer wieder versuchen, sich an neue Trends der Entwicklung anzupassen und sich auch nach den Interessen der SchülerInnen zu orientieren, vorwerfen, die „falsche“ Informatik zu unterrichten.

Mit der „richtigen“ Informatik ist meist das Problemlösen am Computer durch Algorithmen gemeint, ein möglichst rasches Hinführen zum selbstständigen Denken und Handeln am Computer, mit „falscher“ Informatik ist das anwenderorientierte Klicken auf Icons zum Öffnen von Fensterchen gemeint, mit dem man zwar rasch zu guten Ergebnissen kommt, aber von der Idee der „richtigen“ Informatik eben nichts mitbekommt. Eine Schwierigkeit in der Kommunikation zwischen diesen beiden Gruppen liegt oft auch darin, dass wechselseitig von den Interessen des jeweils anderem relativ wenig verstanden wird und es auch keine große Bereitschaft gibt, daran etwas zu ändern, weil der Aufwand dafür aus der jeweils persönlichen Sicht viel zu groß wäre.

Ich selbst besuchte einen damals gerade erstmals neu beginnenden Oberstufenrealgymnasium-Zweig mit Schwerpunkt Informatik. Zusätzlich wurde an unserer Schule über einen Verein die Ausbildung für den ECDL angeboten, die ich schon in der 3. und 4. Klasse der Unterstufe absolvierte. Über diesen Verein konnte man sich auch zum Computerkursleiter ausbilden und so halte ich nun schon seit 12 Jahren Computerkurse, in denen ich vom Volksschulkind bis zum hochbetagten Pensionisten Menschen unterrichte sowie auch Firmenschulungen durchführe. Zusätzlich habe ich auch hunderte Personen außerhalb der Kurse privat betreut, wenn sie Kenntnisse erwerben wollten, die über das Kurscurriculum hinausgingen.

In meiner AHS-Oberstufe wurden verstärkt Programmierkenntnisse vermittelt und zusätzlich richteten wir eine Computerwerkstatt ein, in der wir für die Mitglieder eines extra dafür eingerichteten Computerclubs Reparaturarbeiten und andere Serviceleistungen durchführten.

Nach dieser umfassenden praktischen Erfahrung und meinem parallel dazu absolvierten Lehramtsstudium für Informatik glaube ich nun auch zu wissen, was man sinnvoll im Rahmen des Informatik-Unterrichts an der AHS anbieten könnte und möchte diesen

Vorschlag in einem Konzept für ein Schulbuch zusammenfassen, das sich natürlich auch an die ohnedies sehr allgemein gehaltenen Richtlinien des Lehrplans ganz genau hält. Ich glaube mit diesem Konzept sogar die Lösung gefunden zu haben, von der vor acht Jahren noch geschrieben wurde: „Es wird schwierig sein ein Unterrichtsmodell zu finden, das alle diese Grundsätze ausreichend berücksichtigt.“ Dies haben die Autoren Univ. Prof. Mag. Dr. Karl Fuchs und sein Studienassistent und Diplomand Claudio Landerer über den neuen Informatik-Lehrplan geschrieben.⁸

Ich denke, dass ich ein brauchbares Modell gefunden habe, ich fürchte allerdings, dass es nur wenige Informatik-LehrerInnen derzeit geben wird, die über die Fachkenntnisse verfügen, dieses Modell umzusetzen. Meiner Meinung nach müsste man sich an den Universitäten entschließen, den neuen Trends in der Informatik Rechnung zu tragen, um den StudentInnen eine zeitgemäße Ausbildung zu ermöglichen: Nach meinem Konzept müssten die Schwerpunkte auf PHP, MySQL, Javascript, HTML , XML, CSS und GIMP liegen; all diese Programmier- bzw. Beschreibungssprachen und die Anwendersoftware werden in dem Gratis-Content Management System (CMS) Joomla! benötigt, weshalb ich persönlich mich auf dieses spezialisiert habe. Auch die Arbeit mit Joomla! selbst ist eine eigene Welt, auf die die LehrerInnen vorbereitet werden müssen. Vor allem die Herstellung von barrierefreien Templates und die automatische Anpassung der Websites für Smartphone-Anwender sowie die Programmierung von Apps mit HTML 5 und CSS und alles Organisatorische rund um eine Website-Administration sollten zu den selbstverständlichen Fertigkeiten eines Schülers gehören, der an der AHS einen Schwerpunkt Informatik gewählt hat.⁹

Nun werden natürlich die VertreterInnen der „richtigen“ Informatik sofort einwenden, dass es sich hier um eine unzulässige Beschränkung auf das Webdesign handelt und alle anderen Bereiche der Informatik ausgeklammert werden.

⁸ Helmut Caba und Claudio Landerer: Neuer Informatik-Lehrplan. Was ist neu am neuen Informatik-Lehrplan 5. Klasse AHS? In: CD Austria. 3/2005. S. 11ff. Vgl.

<http://www.gym1.at/schulinformatik/buecher/ahs-informatik.pdf>

⁹ Joomla 3 geht genau in diese Richtung; vgl.:Christoph Prevezanos: Joomla 3. Websites erstellen und gestalten. München: 2013. S. 257ff.

Für mich steht fest, dass ein Schwerpunkt auf technische Informatik (alles in Zusammenhang mit Robotern etc.) in die HTL gehört und ein Schwerpunkt auf die kaufmännische Informatik (Buchhaltung, Kalkulation etc.) in die HAK, in der AHS kann natürlich auch nur ein Ausschnitt aus dem immer breiteren Aufgabengebiet der Informatik geboten werden. Gerade der Bereich rund um das Webdesign deckt alle Anforderungen des Lehrplans ab, wenn man das Projekt richtig verpackt. Das will ich mit meinem Konzept für ein Schulbuch beweisen.

II. Konzept für ein Schulbuch

1. Vorwort

Was soll den LeserInnen in diesem Buch geboten werden? Es ist ein Konzept für einen Informatik-Unterricht an der AHS-Oberstufe, das sich strikt an die Vorgaben des derzeit gültigen Lehrplans hält und doch auf den ersten Blick etwas ungewöhnliche Wege geht.

Die SchülerInnen legen eine Website mit Joomla! an. Dafür erhalten sie eine genaue Schritt-für Schritt-Anleitung, wobei ihnen die Zusammenhänge zwischen Datenbank und Template und der Zusammenhang von PHP, Javascript, HTML, XML und CSS innerhalb des Templates überhaupt noch nicht klar sein müssen. Joomla! ist zunächst eine Blackbox, über die Kategorien zu den einzelnen Unterrichtsfächern erstellt werden und die SchülerInnen den Auftrag erhalten, ab sofort Inhalte zu diesen Unterrichtsfächern auf Joomla! zu veröffentlichen. Sie sollen dabei auch Textverarbeitungs- und Präsentations-Dateien als PDF über FTP hochladen und diese mit den Beiträgen verlinken.

1.1. Was Sie in diesem Buch nicht finden werden

Wenn Sie nun meinen, dass mir hier ein Fehler passiert wäre und man ja zunächst diese Programme in Informatik unterrichten müsste, dann muss ich dies verneinen: Meiner Meinung nach müssen nämlich Textverarbeitung und Präsentation bereits in der Unterstufe durchgenommen werden (egal, in welchem Fach und egal, von welchem Lehrer; das sollten alle LehrerInnen unterrichten können, dazu braucht man keinen Informatik-Lehrer, weil ja auch jeder Lehrer heute mit Textverarbeitungs- und Präsentationsprogrammen selbstverständlich arbeiten können muss). Auch das Versenden von E-Mails und die Arbeit in sozialen Netzwerken (z. B. in Communities von Google+) sollten zur alltäglichen Unterrichtsarbeit der Unterstufe, wenn nicht sogar schon der Volksschule gehören.

Die nächste Enttäuschung wird bei allen KollegInnen folgen, die meinen, man müsste über Hardware im AHS-Informatik-Unterricht sprechen. Auch das halte ich für Schnee von gestern. Die Computer werden immer kleiner und die Zeiten, wo man am sperrigen Desktop-PC noch selbst Hand angelegt hat, sind eigentlich schon längst vorbei. Bald werden Computer schon so klein wie Handys sein und man wird sie lediglich an riesige

Bildschirme anschließen und vielleicht eine externe Tastatur und Maus verwenden. Fünf Jahre Garantie werden zur Regel werden, was bis dahin kaputt geht, wird also gratis repariert und was danach kaputt geht, wird hoffentlich richtig entsorgt...

Die vorletzte Enttäuschung liegt vielleicht in meiner Ablehnung über die Geschichte der Informatik im Informatik-Unterricht zu sprechen. Das ist schon Aufgabe des Geschichtelehrers und ich sehe auch keinen großen Sinn darin über 10 Jahre alte Prozessortechnologien genau zu unterrichten (wie das heute tatsächlich noch gerne auf der Universität gemacht wird). Schade um die Zeit! Einen wichtigen Platz kann das noch auf der HTL haben, auf der AHS sollte man sich auf andere Dinge konzentrieren. Die SchülerInnen sollten Kompetenzen erlernen, die sie nach der Matura im Berufsleben gut brauchen können. Der Erwerb solider Kompetenzen erfordert viel Zeit, da muss man sich auf das Wesentliche konzentrieren und sollte nicht sinnlos abschweifen. Man darf auch nicht vergessen, dass Fertigkeiten oft geübt werden müssen, bis sie wirklich sitzen. Dies muss verstärkt im Unterricht gemacht werden, weil viele SchülerInnen immer weniger bereit sind, zu Hause viel für die Schule zu arbeiten.

Die letzte Enttäuschung liegt in meiner Ablehnung Excel im Informatikunterricht durchzunehmen. Das sollte der Mathematiklehrer in seinen Unterricht einbauen. Die programmierbaren Taschenrechner werden in absehbarer Zeit durch Tablet-PCs ersetzt werden, die natürlich mit Excel am leistungsfähigsten für den Mathematikunterricht sind. Das Gleiche gilt für CAD-Programme, die in Geometrisch Zeichnen oder Darstellender Geometrie, aber natürlich auch in Mathematik eingesetzt werden können. All die Ergebnisse dieser Unterrichtsarbeit sollen in der entsprechenden Kategorie auf der SchülerInnen-Website in einzelnen Beiträgen verlinkt werden, die Arbeiten selbst aber aus dem Informatikunterricht ausgelagert.

Dieses Konzept für den Informatik-Unterricht reicht aus, den Lehrplan zu erfüllen.

1.2. So wird der Informatik-Lehrplan abgedeckt

Der immer noch gültige Lehrplan für die Informatik-Oberstufe von 2005¹⁰ wird derzeit ergänzt durch eine Handreichung zur neuen Matura¹¹. Ich schreibe die Richtlinien des Lehrplans für die 5. Klasse in beliebiger Reihenfolge jeweils in die erste Spalte der nun folgenden Tabelle, in der zweiten Spalte wird vermerkt, wie diese Richtlinien durch mein Konzept erfüllt werden, in die dritte Spalte kommen Anmerkungen.

Tabelle 1: Abdeckung des Informatik Lehrplans durch Joomla:

Richtlinien des Lehrplans	Realisierung durch Joomla!	Anmerkungen
„(...) wesentliche Maßnahmen und rechtliche Grundlagen im Zusammenhang mit Datensicherheit, Datenschutz und Urheberrecht kennen lernen sowie die Auswirkungen des Technikeinsatzes auf die Einzelnen und die Gesellschaft nachvollziehen“ ¹²	Das beginnt schon mit dem Impressum der Website und mit der Sicherung über Akeeba-Backup ¹³ , die sofort nach der Erstellung der noch leeren Website erstmals durchgeführt werden muss. Es ist auch klar, dass auf der Website nur eigenes geistiges Eigentum als eigenes geistiges Eigentum veröffentlicht werden darf.	Dass Unterrichts- und Lernmaterial auf der Website veröffentlicht wird, kann für die ganze Klasse von großem Vorteil sein und diese Vorteile für einen Mikrokosmos lassen sich auch jederzeit für einen Makrokosmos nachvollziehbar machen.
„(...) Informationsmanagement und Lernorganisation für die eigene Lernarbeit und Weiterbildung mit geeigneter Software in der Praxis umsetzen und dabei vorhandene Informationsquelle erschließen und unterschiedliche	Da die SchülerInnen auf der Website Unterrichtsmaterial zu allen Gegenständen in den richtigen Kategorien unter nachvollziehbaren Kriterien abspeichern und aufbereiten müssen (auch unter Einbeziehung von PDF-Dateien aus Textverarbeitungs- und Präsentationsprogrammen) wird auch dieser Punkt erfüllt.	Selbstverständlich können immer auch weitere Informationsquellen zum Thema verlinkt werden.

¹⁰ Vgl. http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf

¹¹ Vgl. http://www.bmukk.gv.at/medienpool/22837/reifepruefung_ahs_mrp.pdf

¹² Vgl. http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf

¹³ Vgl. <http://www.akeebabackup.com/>

Informationsdarstellungen ausgehend von den Vorkenntnissen anwenden“ ¹⁴		
„(...) Inhalte systematisieren und strukturieren sowie Arbeitsergebnisse zusammenstellen und multimedial präsentieren können“ ¹⁵	Hier wird auch das Smartphone eine große Rolle spielen, können doch auch Fotos und Videos zum Thema beispielsweise über Google+ rasch veröffentlicht und entsprechend verlinkt werden (auf Wunsch auch für einen sehr begrenzten User-Kreis).	In der gerade aktuellen Diskussion, ob das Handy in der Schule vielleicht sogar verboten werden sollte, muss ich aus meiner Sicht klar anmerken, dass der selbstverständliche Umgang mit dem Smartphone ein wichtiges Lernziel an der Schule sein sollte. Genauso wie der erzieherische Aspekt, wann und in welchen Situationen man zum Smartphone greifen darf und wann eben nicht (das ist auch vielen Erwachsenen derzeit überhaupt nicht klar).
„(...) ein vernetztes Informationssystem für die individuelle Arbeit aufbauen und nutzen können“ ¹⁶	Dies wird durch die Websites der SchülerInnen perfekt erfüllt, da innerhalb der Klasse alle Websites untereinander zugänglich sein sollten und somit eine Vernetzung aller Informationssysteme durchgeführt ist.	Auf diese Weise werden die SchülerInnen schon sehr früh (ab der 9. Schulstufe) dazu erzogen, miteinander zu kooperieren. In der Regel passiert das erst – wenn auch nur teilweise – bei der Maturavorbereitung.
„(...) den sicheren Umgang mit Standardsoftware zur schriftlichen Korrespondenz,	Die einfache Arbeit mit Joomla! (Artikel anlegen, Bilder, Videos und PDF-	Natürlich ist es völlig gleichgültig, mit welchem Sozialen Medium die

¹⁴ Vgl. http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf

¹⁵ Vgl. http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf

¹⁶ Vgl. http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf

zur Dokumentation, zur Publikation von Arbeiten, zur multimedialen Präsentation sowie zur Kommunikation erreichen“ ¹⁷	Dateien verlinken) sollte bald zur Routine werden. Die Kommunikation unter den SchülerInnen sollte per Computer und Handy nicht nur über E-Mail, sondern auch über diverse Communities erfolgen, die man beispielsweise mit Google+ anlegen kann.	SchülerInnen SINNVOLL arbeiten lernen (vielleicht wird ja gerade eines erfunden, während ich hier schreibe, das in zehn Jahren Marktführer ist).
„(...) Kalkulationsmodelle erstellen und die Ergebnisse bewerten und interpretieren können; eine einfache Datenbank benutzen können“ ¹⁸	Wie schon oben erwähnt, überlasse ich die Kalkulationsmodelle dem Mathematikunterricht. Auch in Deutsch ist in der 5. Klasse vorgesehen, dass die SchülerInnen zu Diagrammen frei sprechen lernen und diese sinnvoll interpretieren können. ¹⁹ Datenbanken werden schon mit Joomla! benutzt (gleich über 60!). Darauf kann man mit der Zeit hinweisen, aber auch die Erstellung einfacher Datenbanken in MySQL und parallel in Access beispielsweise üben.	Die Funktionalität von Datenbanken wird vor allem über SQL deutlich. Egal, ob in Joomla!-Datenbanken oder in Access-Datenbanken, man kann einfache Anwendungen dieser Sprache immer wieder üben. Beim Arbeiten mit Formularen wird dann die Aufgabe von SQL besonders klar ersichtlich.
„(...) Einblicke in wesentliche Begriffe und Methoden der Informatik, ihre typischen Denk- und Arbeitsweisen, ihre historische Entwicklung	All dies kann man einfach besprechen, wenn man die Blackbox Joomla! aufschnürt und zeigt, wie die PHP-Dateien über SQL	In weiterer Folge kann man beginnen einfache Plugins, aufwändigere Module oder auch komplexe Komponenten in ihrer

¹⁷ Vgl. http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf

¹⁸ Vgl. http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf

¹⁹ Vgl. Gertraud Hilger, Maria Kiener: klar text Deutsch5. Wien: 2010. . S. 115f.(Schulbuch Nr. 140586)

sowie ihre technischen und theoretischen Grundlagen gewinnen und Grundprinzipien von Automaten, Algorithmen und Programmen kennen lernen“ ²⁰	auf die Datenbank zugreifen, wie HTML- und CSS-Dateien für die Gestaltung des Templates zuständig sind und warum man so sehr darauf achten sollte, barrierefreie Templates zu erstellen.	Funktionsweise zu verstehen, nach- oder auch neu zu programmieren. Wie schon oben erwähnt, würde ich die historische Entwicklung in den Geschichtsunterricht auslagern. Für solche Themen ist bei diesem doch recht arbeitsintensiven Konzept einfach keine Zeit.
Der Lehrplan sieht natürlich für die 6.-8. Klasse noch weitere Vertiefungen vor, sei es für Wahlpflichtfächer oder typenbildende Schwerpunktfächer: „Der Unterrichtsgegenstand Informatik leistet durch die drei Dimensionen Wissensentwicklung, Unterrichtsmanagement und Wissensdarstellung einen unverzichtbaren Beitrag zur Allgemeinbildung. Der Unterricht ist auf der Basis dieser Grundsätze kontinuierlich zu planen und durchzuführen sowie laufend zu reflektieren und anzupassen.“ ²¹	Es ist offensichtlich, dass „Wissensentwicklung, Unterrichtsmanagement und Wissensdarstellung“ durch ein CMS, das alle Unterrichtsfächer strukturiert darstellen, vertiefen und veranschaulichen soll, bestens umgesetzt werden.	Die Herausforderung für den Informatiklehrer ist hier nicht gering, da er auch die anderen KlassenlehrerInnen überzeugen müsste, dass sie diese Konzept unterstützen sollen. Umso breiter die Wissensdarstellung auf der Website ist, umso anspruchsvoller wird das Konzept.
Die Themen sind dabei so auszuwählen, dass sie vielseitige Bezüge aus der	Auch in diesem Zusammenhang muss man fast annehmen, dass die	Diese Idee ist auch gar nicht so weit hergeholt, ist doch schon seit Jahren klar, dass

²⁰ Vgl. http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf

²¹ Vgl. http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf

Lebens- und Begriffswelt der Jugendlichen aufgreifen. (...)Das Unterrichtsmanagement hat sich an für Schülerinnen und Schüler transparenten Lehrzielen zu orientieren und soll beispielgebend für die eigene Lern-und Arbeitsorganisation auch außerhalb des Informatikunterrichts sein. Variierende Arbeitsformen wie Einzelarbeit, Gruppenarbeit und Teamarbeit geben Schülerinnen und Schülern Gelegenheit, Neues zu erforschen und bereits Gelerntes in verschiedenen kommunikativen und inhaltlichen Kontexten anzuwenden.²²	Verantwortlichen für den Lehrplan ein CMS im Hinterkopf gehabt haben müssen, das gemeinsam mit einem sozialen Netzwerk wie Google+ von den SchülerInnen für die Unterrichtsarbeit genützt werden soll.	die wesentliche Entwicklung der Informatik in dieser Richtung verläuft. Die weltweit spürbaren Innovationen haben sich im Bereich des Internets und der Telefonie, also im Kommunikationsbereich, entwickelt.
---	---	--

Auch wenn weiter unten von der „Bedeutung gemeinschaftlichen Problemlösens“ geschrieben wird oder von der „Erweiterung der Sichtweise, Orientierung und Vertiefung von Wissen und Können“ geschrieben wird, passt dies bestens ins Konzept, können sich doch SchülerInnen nach Interessen und Begabungen selbstständig für eigene Website-Projekte zusammenschließen.

Auch wenn es darum geht sich „Kritik anderer zu stellen und ihre Arbeit zu argumentieren“ ist die Website ein hervorragendes Mittel, weil die eigene Arbeit öffentlich ist und daher auch diskutiert bzw. einfach nur kommentiert werden kann.

²² Vgl. http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf

1.3. Was Sie in diesem Buch finden werden

Nach der umfangreichen Beweisführung der Lehrplankonformität soll nun kurz dargestellt werden, was tatsächlich Inhalt dieses Buches sein soll.

In diesem Buch wird der Unterrichtsverlauf klar strukturiert und argumentiert. Es wird beispielsweise keine Anleitungen gegeben, wie etwas technisch zu machen ist (z. B. die Installation von Joomla!). Dafür verweise ich auf Fachliteratur und Weblinks, da in diesem Buch nicht eine Zusammenfassung bereits oft veröffentlichter Inhalte angeboten werden soll. Im Buch werden also eher die die Zeit überdauernden Inhalte angeboten, dadurch wird es nicht so rasch veralten wie herkömmliche Informatikbücher. Die technischen Anleitungen auf guten Websites werden laufend aktualisiert, die Links bleiben aber meist gleich.

Folgende Lernziele sollen durch die einzelnen Kapitel des Buches erreicht werden:

- a) Wissen, was ein CMS ist und Installation von Joomla! auf einem Gratisserver (z. B. <http://bplaced.net> oder <http://square7.ch>) Genaue Anleitung dazu z. B. auf <http://eass.bplaced.net/141-Joomla!-30>
- b) Begreifen, was eine gute Struktur eines CMS ist. Vorschlag für eine Struktur. Kritisches Hinterfragen einer CMS-Struktur lernen. Genaue Anleitung, wie man eine Struktur sinnvoll anlegt.
- c) Den Unterschied zwischen Inhalt (Datenbank) und Layout (Template) einer Website verstehen. Unterschiedliche Templates installieren und aktivieren. Auf gute Anleitungen dazu verweise ich mit entsprechenden Links.
- d) Lernen, wie man Templates beurteilen kann; Auflistung kritischer Fragen. Die Bedeutung von barrierefreien Templates begreifen. Genaue Erklärung, wie ein Template aufgebaut ist und wie man eines selbst erstellt, kann man wieder extern nachlesen. Gute Verweise werde ich anbieten.
- e) Die Bedeutung des Smartphones im Unterricht: Alle wesentlichen Funktionen sollen beherrscht werden, von der Terminverwaltung über die Notizen bis zum Upload von Fotos und Videos. Über Windows 8 werden die Smartphones immer mehr zu Kleincomputern, die man mit Tablets und Notebooks (herkömmliche Notebooks werden bald aussterben und es wird nur noch Notebooks geben, die man auch als Tablets nutzen kann) bestens synchronisieren kann. Vermutlich

werden sich auch nicht die in den Bildschirm eingebauten Computer durchsetzen, sondern man wird Handys oder Tablets an große Bildschirme anschließen. Auch über diese Entwicklungen könnten LehrerInnen und SchülerInnen in einer Google+-Community Erfahrungen austauschen.

- f) Grundlagen der PHP-Programmierung sollen verstanden werden. Einfache und komplexe Übungsaufgaben werden von mir zum freien Download zur Verfügung stehen.
- g) Grundlagen der SQL-Programmierung sollen verstanden werden. Einfache und komplexe Übungsaufgaben werden von mir zum freien Download zur Verfügung stehen. Dabei soll auch bewusst gemacht werden, dass nicht nur MySQL, sondern auch Access in SQL geschrieben ist. Die Grundlagen der Datenbankprogrammierung sollen begriffen werden, um sie in einfachen Beispielen selbst anzuwenden (z. B. Abfragen).
- h) Grundlagen der CSS-Programmierung sollen verstanden werden. Einfache und komplexe Übungsaufgaben werden von mir zum freien Download zur Verfügung stehen. Besonders das Zusammenspiel von HTML und CSS muss durchschaut werden.
- i) JavaScript-Programmierung findet bei Joomla! immer wieder Verwendung. An einigen Beispielen soll die besondere Funktionalität von JavaScript durchschaut werden und es soll gelingen, dass bestehende Programme abgewandelt oder neue Programme geschrieben werden können. Einfache und komplexe Übungsaufgaben werden von mir zum freien Download zur Verfügung stehen.
- j) Die Grundlagen der Suchmaschinenoptimierung sollen verstanden und angewendet werden können.

Wie man sieht, ist dies alles ein ambitioniertes Unternehmen für den Zeitraum von vier Jahren. Wozu braucht man dafür dieses Lehrbuch, stehen nicht alle Informationen zu diesen Themen in unzähligen Büchern der jeweils speziellen Informatikliteratur?

Das ist richtig, aber man erspart sich als Lehrer die Lektüre all dieser Bücher, wenn man nach meinem Leitfaden vorgeht. Außerdem verfolge ich auch ein besonderes Prinzip im Aufbau der zu vermittelnden Inhalte. Vieles soll GLEICHZEITIG vermittelt werden. Wenn man zuerst HTML und PHP macht, dann SQL und CSS, haben die SchülerInnen bei CSS

schon wieder vergessen, wie HTML funktioniert, was aber wichtig zu wissen ist, wenn man über CSS-Selektoren auf HTML-Strukturen zugreifen will. So muss man sich bei den einzelnen Kapiteln durch spezielle Übungen immer wieder auf die bereits durchgenommenen Kapitel beziehen. Viel zu tun!

Sollten sich allerdings andere LehrerInnen dieser Idee anschließen und eigene Module zur Unterrichtsgestaltung ebenfalls veröffentlichen, wird das Schwierige plötzlich einfach, weil ein produktiver Austausch zwischen den LehrerInnen stattfinden könnte und damit genau das Gleiche, was man mit den SchülerInnen in der Klasse erreichen will: eine gute Kultur der Zusammenarbeit.

In den folgenden Abschnitten werde ich einige Unterrichtsmodule skizzieren, bei denen es um die Einrichtung von Joomla! und der Planung eines CMS geht.

2. Joomla! im Unterricht:

2.1. Einfache Unterrichtseinheiten:

Hier präsentiere ich einige kurze Unterrichtsbeispiele, um zu zeigen, wie ich mir das vernetzte Arbeiten mit Joomla!, HTML, CSS, PHP und SQL vorstelle. Es geht dabei darum, zunächst ganz an der Oberfläche zu bleiben und nur die notwendigsten Grundstrukturen zu erläutern, um dann immer mehr in die Tiefe zu gehen. Diese Beispiele zeigen Übungen an der Oberfläche. Wichtig ist für mich, dass man von Anfang an ALLE Aufgabenbereiche einführt, um dann parallel jeweils immer mehr zu vertiefen, aber auch zu wiederholen, um zu sichern, dass die Grundlagen gut beherrscht werden, auf denen dann ja weitere Kompetenzen aufgebaut werden sollen. Die SchülerInnen sollen jedenfalls mit Kompetenzen aus dem Unterricht gehen, die ihnen im späteren Berufsleben auf alle Fälle nützlich sein können. Und: Sie sollen diese Kompetenzen tatsächlich HABEN!

2.1.1. Einrichtung von Joomla!:

Joomla! ist ein Content Management System (CMS), mit dem man – wie der Name schon sagt – Inhalt verwalten kann. In unserem Fall soll es der Stoff sein, der in den unterschiedlichen Fächern unterrichtet wird. Bevor man den Stoff im CMS

veröffentlichen kann, muss man das CMS installieren. Wir wählen Joomla!, das man zunächst als Paket von <http://www.Joomla!.os.de> herunterladen muss. Dieses Paket wird entpackt und danach über FTP (wir wählen das FTP-Programm FileZilla, Download: <http://www.filezilla.de/download.htm>) wieder auf einen Server gespielt (nehmen Sie z. B. <http://bplaced.net> oder <http://square7.ch>). Um über FTP Daten auf einen dieser Server übertragen zu können, muss man dort ein Benutzerkonto anlegen. Eine genaue Anleitung dafür gibt es auf z. B. auf <http://eass.bplaced.net/45-Joomla!>

Sind die Daten übertragen, ruft man den Ordner auf, in dem man Joomla! abgespeichert hat und die Installation beginnt. Da in Joomla! der Inhalt immer in einer Datenbank organisiert wird, muss man vor der Installation auch eine Datenbank anlegen.

Bei der Joomla!-Installation muss man dann den Namen der Datenbank und den Benutzernamen für die Datenbank eingeben (standardmäßig lauten die bei unseren beiden Gratis-Providern gleich) sowie das für die Datenbank gewählte Passwort.

Die SchülerInnen lernen also schon bei der Installation, wie man mit einem FTP-Programm arbeitet, wie man einen Benutzer bei einem Provider anlegt, wie man dort eine Datenbank erstellt und wie man Joomla! bei der Installation mit eben dieser Datenbank verbindet.

Wo liegen die Vorteile eines CSM? Inhalt und Layout sind getrennt zu bearbeiten; das Layout der Website wird in Joomla! als Template (engl. Schablone) bezeichnet und kann unabhängig vom Inhalt in eine beliebige Form gebracht werden; umgekehrt kann auch der in einer Datenbank organisierte Inhalt der Website nach Belieben exportiert werden, um ihn in eine andere Website einzufügen.

Mehr muss man zu diesem Thema für den Anfang nicht wissen. Im nächsten Abschnitt geht es nun darum, wie man den Inhalt auf der Website sinnvoll strukturiert.

2.1.2. Inhaltliche Struktur eines CMS:

Ähnlich wie bei einer Datenbank muss man vor der Erstellung eines CMS sich einmal zurücklehnen und genau überlegen, wie die Inhalte strukturiert sein sollen. Dabei können auch Bleistift und Papier sinnvoll sein, um rasch einen groben Aufbau der Website zu skizzieren. Natürlich gibt es auch dafür schon Computerprogramme, aber in

diesem Fall ist diese altbewährte Methode vielleicht noch am zeitsparendsten. Da der Anfänger gerne wichtige inhaltliche Punkte übersieht, machen wir hier einen Vorschlag für eine Grobstruktur. Webdesigner sollten sich auch angewöhnen, fremde Websites mit kritischem Blick zu betrachten.

Wichtig ist, dass eine Website, die laufend mit Inhalten erweitert wird, so aufgebaut ist, dass sie nach Belieben wachsen kann und dabei übersichtlich bleibt. So schlage ich die einzelnen Unterrichtsfächer als Kategorien vor und als Unterkategorien die Monate des Schuljahres. So bekommt man einen raschen Überblick, was wann in den jeweiligen Fächern veröffentlicht wurde. Eine genaue Anleitung, wie man Kategorien, Unterkategorien und Beiträge erstellt, findet man auf unzähligen Websites. Besonders wichtig dabei ist auch noch, dass die SchülerInnen lernen Textverarbeitungs- und Präsentations-Dateien als PDF abzuspeichern, über FTP hochzuladen und dann mit einem Joomla!-Beitrag zu verlinken. Am besten gelingt dies, wenn man die hochgeladene Datei im Browser aufruft und die Adresse kopiert um sie als Link in Joomla! einzufügen.

Beiträge auf Webseiten sind oft dann besonders beliebt, wenn sie auch Bilder und Videos beinhalten. Beides ist über Google+ sehr einfach. Google bietet ja auch die Bildbearbeitung Picasa an, mit der Bilder nicht nur bearbeiten, sondern auch sehr einfach ordnen und veröffentlichen kann. Auf <https://support.google.com/picasa/?hl=de> gibt es dazu sehr gute Anleitungen; damit SchülerInnen rasch Bilder veröffentlichen und mit dem jeweiligen Beitrag verlinken können, genauso Videos, die man auf <http://www.youtube.com> ebenfalls sehr einfach über Google+ (Goggle hat ja auch Youtube gekauft) veröffentlichen kann und diese veröffentlichten Videos verlinkt man dann an einer beliebigen Stelle und auch ab einem bestimmten Sendezeitpunkt mit einem Joomla!-Beitrag. Selbstverständlich kann man auch sehr einfach einzelne Bilder in Beiträge einfügen und diese per Anklicken dann in einer vergrößerten Form darstellen, wenn man das kleine Bild mit dem größeren, ebenfalls über FTP hochgeladenen verlinkt. Die Erstellung von Beiträgen wird für viele SchülerInnen zunächst eine große Herausforderung sein: Wenn Sie selbst Fotos oder Videos machen (z. B. von Versuchen in Physik oder Chemie, Pflanzen oder Tiere in Biologie, Zeichnungen oder Werkstücke usw.) wird man in vielen Fällen bemerken, dass die SchülerInnen keine guten Fotos

machen können. Das liegt nicht an den Handy-Kameras (die werden immer leistungsfähiger), sondern einfach an dem Unvermögen zu erkennen, woher das Licht kommen soll und welcher Bildausschnitt im jeweiligen Fall vernünftiger Weise gewählt werden soll. Noch schwieriger wird die Sache bei der Herstellung von Videos; egal, ob es sich um ein Interview handelt oder um die Beobachtung eines bestimmten Vorgangs im Sinne einer Dokumentation. Hier sind vielfach nicht nur die SchülerInnen gefordert, sondern auch die LehrerInnen und es empfiehlt sich sehr, dass gerade die LehrerInnen hier hilfreich eingreifen, die in diesen Bereichen ein gewisses Talent haben.

Ziel dieser Unterrichtseinheit ist es jedenfalls, dass die SchülerInnen Beiträge in beliebigen Kategorien und Unterkategorien anlegen können und die Inhalte mit Fotos und Videos auch besser veranschaulichen können. Die Qualität der Beiträge könnte von den Fachkollegen überprüft werden und es gäbe natürlich auch die Möglichkeit die Website mehrsprachig anzulegen (auch dafür bietet Joomla! ein einfaches Modul, das praktisch selbsterklärend ist), wenn man auch die Fremdsprachen in dieses Projekt stärker einbinden möchte.

Manche SchülerInnen werden einige Wochen Zeit brauchen, bis sie wirklich routiniert Beiträge anlegen, richtig einordnen und mit Bildern und Videos veranschaulichen können, es reicht aber bei regelmäßiger Übung bestimmt ein Schuljahr aus, um in diesem Bereich eine lebenslange Kompetenz zu erlangen. All das hat schon viel mit dem zu tun, was im Informatik-Lehrplan gefordert wird, es hat aber noch gar nichts mit dem zu tun, was von vielen LehrerInnen gerne als „richtige“ Informatik bezeichnet wird. Auch dazu sollte man aber schon relativ rasch in der 5. Klasse kommen, da viele SchülerInnen gerade in diesem Bereich viele Wiederholungen benötigen, bis wichtige Grundlagen wirklich sitzen, auf denen man dann den weiteren Unterricht aufbauen kann. Bei einem kürzlich von mir gehaltenen Unterrichtspraktikum am Gymnasium St. Ursula fiel mir allerdings auf, dass die meisten SchülerInnen sich sehr rasch mit Joomla! zurechtfinden.

2.1.3. Die Blackbox Joomla! wird zerlegt:

Ist man nun soweit, dass man mit Joomla! arbeiten kann, dann sollte man auch schon daran gehen, die einzelnen Teile von Joomla! zu begreifen, indem man sie sich genau anschaut.

2.1.3.1. Templates, Menüs und andere Erweiterungen oberflächlich betrachtet:

Nehmen wir zunächst das Template: In einer groben Struktur zur Erstellung von Inhalt können wir in Joomla! das Menü oben und das Hauptmenü links lassen und alle anderen vorinstallierten Menüs einfach löschen. Will man nun die Arbeit mit Joomla! verfeinern, muss man zunächst die Bedeutung von Plugins, Modulen und Komponenten verstehen und dann in der Lage sein, eigene Menüs zu erstellen, diese in Module einzubinden und die Module dann an einer bestimmten Stelle des Templates einzubinden und sichtbar zu machen. Auch dazu findet man im Internet genaue Anleitungen.

Auf diesem Weg lernt man die Grundbausteine (Plugins, Module und Komponenten) von Joomla! kennen und verliert die Scheu, mit diesen ein wenig zu experimentieren. Auch hier sind wir noch immer im Bereich der einfachen „Klick-Informatik“. Es tut sich aber schon jetzt eine faszinierende Welt auf, wenn man bereits vorhandene Bausteine verändert, damit genau die Funktionen realisiert wird, die man eben von diesem Baustein wünscht. Dann sind wir bei der „richtigen“ Informatik, wo es ein Problem gibt und wo ein Algorithmus gefragt ist, der dieses Problem lösen kann. Das Gute an diesem Unterrichtsmodell ist, dass man mit der ganzen Website beginnt und sich aber relativ bald Fragen nach Details stellen, die aber erst beantwortet werden können, wenn PHP, HTML, CSS, SQL und JavaScript einiger Maßen beherrscht werden. So entsteht hoffentlich der Wunsch in möglichst vielen SchülerInnen, genau diese Kompetenzen zu beherrschen. Der Lehrer könnte zu diesem Zeitpunkt sehr gefordert sein, da er in all diesen Bereichen Expertenwissen benötigt, um den Wünschen der SchülerInnen gerecht zu werden. Zur Zeit wäre das ein schwieriges Unterfangen, da kaum einer dieser Bereiche im Rahmen der LehrerInnen-Ausbildung vermittelt wird...

Der erste Weg ist immer, nach schon gefertigten Bausteinen im Internet zu suchen. Joomla! ist eine Open-Source-Software, man kann also alles, was bereits erstellt wurde, nach Belieben verändern oder eben einfach so verwenden, wie es programmiert wurde.

Ein erster sinnvoller Weg ist bestimmt der, dass man seine Website mit bereits bestehenden Bausteinen erweitert, egal, ob es sich um ein Gästebuch, eine Fotogalerie, ein Formular oder eben ein Template handelt. All diese Bausteine findet man auf unzähligen Websites übersichtlich geordnet und oft auch in einem Beliebtheitsranking.²³

Manche werden mit dem Gebotenen zufrieden sein, andere werden beginnen, an den vorhandenen Bausteinen zu basteln und so ergibt sich im Informatik-Unterricht eine automatische Differenzierung, wo sich die einen mehr auf den Content konzentrieren werden und die anderen auf die Form, in der der Content präsentiert werden könnte.

Um Veränderungen durchführen zu können, müssen allerdings gute HTML-, CSS- und PHP-Kenntnisse vorhanden sein. Dazu kommen wir erst später...

2.1.3.2. Datenbanken kurz angeschaut...

Ein weiterer Teil der Blackbox ist die Datenbank. Ein erster Schritt ist einfach der, sich über phpMyAdmin einmal die Datenbank anzuschauen, die man erstellt hat, um Joomla! zu installieren. Verblüfft stellt man fest, dass dort durch die Installation über 60 Tabellen entstanden sind. An dieser Stelle muss man sich klar machen, wie eine Datenbank aufgebaut ist: Die Datenbank selbst ist die Hülle für die Tabellen, die man in einer Datenbank nach Belieben erstellen kann. Die Tabellen werden über den Primärschlüssel (eine beliebte Abkürzung ist id = Identifier) miteinander verbunden, indem der Primärschlüssel einer Tabelle in einer anderen Tabelle als sogenannter Fremdschlüssel eingefügt wird. Die Tabellen bestehen wiederum aus Datensätzen, das ist der Fall in einer Tabelle (wenn ich also eine Tabelle mit den Adressen meiner Freunde anlege, muss ich dieser Tabelle einen Namen geben, z. B. adressen_freunde, muss mir Datenfelder überlegen, z. B. Vorname, Name, Straße [man darf keine Sonderzeichen in Datenfeldnamen verwenden, muss also Strasse schreiben], PLZ ..., und jeder Freund, den ich dann in die Datenbank in die entsprechende Tabelle eintrage, ist ein Fall).

Soviel zur grauen Theorie. Datenbanken begreift man allerdings viele besser, wenn man gleich damit arbeitet. Es gibt hier ganz unterschiedliche Zugänge: Holger Reibold lässt

²³ Vgl. z. B. <http://www.Joomlaos.de>

gleich zu Beginn seines Buches eine Joomla!-Komponente erstellen²⁴, Johann-Christian Hanke wagt sich erst gegen Ende seines Buches an eine Datenbank, um die Bestellung von Kinokarten unter Freunden zu organisieren.²⁵

Ich glaube, dass es für den Anfang reicht, wenn man zunächst eine neue Datenbank für Übungszwecke anlegt (das sollte man schon von der Joomla!-Installation kennen) und dann in dieser Datenbank eine Tabelle anlegt. Zuerst sollte man das über die Funktion „Tabelle erstellen“ machen und dann das Gleiche viel einfacher über die Code-Eingabe im SQL-Fenster. Am einfachsten ist es, wenn man sich den SQL-Code downloadet und entsprechend adaptiert. Die nächste Herausforderung ergibt sich dann mit ganz einfachen Abfragen, die man mit zunehmender Sicherheit dann auch immer komplexer gestalten kann. Die SchülerInnen können hier frei experimentieren und sehen an der Ausgabe sofort, ob die Abfrage erfolgreich war. Zu diesem Zeitpunkt kann man vielleicht ein wenig in phpMyAdmin arbeiten und ein paar Abfragen starten, man hat aber noch nicht wirklich die Struktur einer Datenbank verstanden. Das ist erst möglich, wenn man sich eine auf die Realität bezogene Aufgabe stellt: z. B. als Besitzer eines Autohauses, das Reparaturen und Verkauf anbietet, will man darüber eine Datenbank erstellen. Welche Tabellen benötigt man und wie müssen diese miteinander verbunden sein? Dafür ist es aber jetzt noch zu früh. Vorläufig reicht das einfache technische Üben mit der Materie. Wir befinden uns zu diesem Zeitpunkt also immer noch im Bereich der „Klick-Informatik“, arbeiten mit Versuch und Irrtum...

2.1.4. Das Zusammenspiel von HTML und CSS

Es ist von Anfang an ein breiter Bogen zu spannen. Werfen wir nun einen kurzen Blick auf HTML²⁶ und CSS.

Jede Webseite im Internet wird mit der Beschreibungssprache HTML definiert (Website ist im Gegensatz zur Webseite der gesamte Internet-Auftritt, der unter einer bestimmten Adresse veröffentlicht wird). HTML ist für Struktur und Inhalt der Webseite zuständig, CSS für die Formatierung und das Layout. Auch diese Aussage wird für die

²⁴ Holger Reibold: phpMyAdmin kompakt. Saarbrücken: 2009. S.11ff. (Früher <http://www.bomots.de>, jetzt: <http://www.brain-media.de>)

²⁵ Johann-Christian Hanke: PHP und MySQL für Kids. Heidelberg u.a. 2010. S. 391ff.

²⁶ HTML-Editoren gibt es viele und es ist Geschmackssache, welche man nun tatsächlich verwenden will. Ich arbeite schon lange mit PSPAD

SchülerInnen auf den ersten Blick abstrakt wirken. Schauen wir uns zuerst die Struktur einer HTML-Seite an. Am besten lädt man sich eine Grundstruktur einer HTML5-Webseite aus dem Internet und schaut sich diese in Ruhe an.

Am Beginn steht Code für die allgemeine Beschreibung, dann folgen der head und schließlich der body. Im head kann es einen Link (also eine Verbindung) zu einer CSS-Seite geben, die für die Formatierung und das Layout der HTML-Seite zuständig ist. Auch diesen Zusammenhang versteht man am besten, wenn man das Ganze einfach ausprobiert. Man erstellt eine einfache HTML-Seite `html.html` und die Übungsseite für die CSS-Seite `css.css`. Den Code für beide Dateien stelle ich als Download zur Verfügung. Wenn man nun beide Dateien auf einen Gratis-Server über das FTP-Programm im selben Ordner abspeichert, werden die in der CSS-Datei definierten Formatierungen auf der HTML-Seite sichtbar. Die Funktionsweise ist immer gleich: Über den Selektor greift die CSS-Datei auf einen bestimmten strukturierten Bereich der HTML Seite zu und dann stehen in der geschwungenen Klammer (dem sogenannten Deklarationsblock) die Deklarationen, die aus Eigenschaft und Wert bestehen, die durch einen Doppelpunkt getrennt sind und von einem Strichpunkt abgeschlossen werden. Wenn man das verstanden hat, kann man jeden beliebigen Bereich der HTML-Seite über die CSS-Seite formatieren. Das ist natürlich nur in den Grundstrukturen einfach und wird in der Folge viel komplizierter (wenn es zum Beispiel um das Vererben geht...). Jetzt geht es aber einmal nur darum, dass man diesen ganz einfachen Schritt beherrscht. Rasch bekommt man Lust, etwas mehr in HTML und CSS zu versuchen. Dafür kann man sich unzählige Code-Beispiele in HTML²⁷ und CSS²⁸ aus dem Internet laden und diese ausprobieren. Eine gute Möglichkeit ist auch mit dem Browser Google-Chrome zu surfen und bei einer Website, die man gerne näher untersuchen will, mit der rechten Maustaste auf den Bildschirm zu klicken und dann im Kontext-Menü auf Element untersuchen. Dies ist aber nur für die Vorwitzigen gedacht. Im Prinzip reicht es zu diesem Zeitpunkt, wenn man eine einfache HTML-Seite erstellen kann, diese mit einer CSS-Seite verbinden kann, auf der CSS-Seite ein paar Deklarationen definiert, über die einige Strukturen der HTML-

²⁷ Siehe z. B. <http://de.selfhtml.org/> (selten verdient etwas mehr die Bezeichnung alt, aber gut...)

²⁸ Siehe z. B. <http://de.selfhtml.org/css/intro.htm> oder <http://www.csszengarden.com/tr/deutsch/> Ein wirklich empfehlenswertes Buch über CSS: David Sigos: CSS für Kids. Heidelberg: 2012.

Seite formatiert bzw. layoutiert werden. Nicht oft genug sollte man wiederholen: CSS ist für Formatierung und Layout zuständig, HTML für Inhalt und Struktur.

2.1.5. Weg von der „Klick-Informatik“: PHP

Bis jetzt war alles einfach zu verstehen und das kann natürlich nicht immer so bleiben.

Die Informatik hat auch ihre Hürden, die wir jetzt langsam versuchen werden zu nehmen. Sind HTML und CSS Beschreibungssprachen, die für Inhalt und Aussehen einer Website zuständig sind, beginnen wir jetzt mit einer richtigen Programmiersprache, die serverseitig funktioniert, das heißt, alle PHP-Programmierung, die nicht in einer lokalen Serversimulationsumgebung (z. B. XAMPP²⁹) oder auf einem echten externen Server, der PHP-tauglich ist (z. B. <http://bplaced.net>, <http://square7.ch>), gespeichert wird, funktioniert gar nicht. Im Gegensatz dazu läuft Javascript, die zweite Programmiersprache, mit der wir uns in Zusammenhang mit Joomla! befassen müssen, clientseitig. Dazu kommen wir aber später.

Wie nähert man sich einer Programmiersprache an? In der Regel wird zunächst immer die berühmte Ausgabe „Hallo Welt“ erzeugt. Dann wagt man sich stufenweise an komplexere Programmiervorhaben.

Für unsere Zwecke ist ein anderer Ansatz sinnvoller: Wir versuchen die index.php eines Templates zu analysieren, um mehr von PHP zu begreifen.

Dazu verwenden wir die index.php eines sehr beliebten Templates namens Allrounder³⁰.

Eine genaue Beschreibung dieser Datei findet man auf

<http://www.lernvid.com/downloads/allrounder-Joomla!-16-template.html>. An dieser Stelle sollen nur ein paar allgemeine Hinweise stehen:

Eine php-Datei ist eine html-Datei, in der auch php-Code enthalten ist. Dieser Code wird in der Regel durch `<?php` eingeleitet und durch `?>` abgeschlossen.

Es können in einer html-Datei beliebig viele php-Einschübe sein, wie man an der vorliegenden index.php sieht. Sie beginnt sogar schon mit php-Code, danach folgt die schon besprochene übliche html-Struktur. Was der jeweilige php-Code bewirkt, kann

²⁹ Vgl. <http://www.apachefriends.org/de/xampp.html>

³⁰ <http://www.lernvid.com/downloads/allrounder-Joomla!-16-template.html>

man googeln³¹ oder auch aus einer großen Zahl von Büchern wählen.³² Für uns genügt es vorerst zu beachten, dass eine große Zahl von DIV-Containern definiert wurde, in die immer wieder auch PHP-Code eingebaut wurde, der sich konkret auf Joomla! bezieht, das ja auch in PHP geschrieben wurde. Hier geht es um die Positionierung der Module, in denen wieder eigene Menüs oder andere Joomla!-Funktionen hineingelegt wurden.

Da das Verständnis für diese Datei eine Grundvoraussetzung für das Verständnis von Joomla! ist, wollen wir uns über unterschiedliche Wege annähern, die Datei sozusagen umkreisen und unter verschiedenen Aspekten betrachten.

Offensichtlich kann man PHP nicht ohne HTML verstehen. Wie funktioniert das mit den DIV-Containern. Man könnte es theoretisch beschreiben, wir ziehen aber wieder eine praktische Übung vor. Danach kommen wir wieder zu PHP zurück...

2.1.6. Wie funktioniert das mit den DIV-Containern in HTML?

Ich habe schon auf eine meiner Lieblingsseiten im Netz hingewiesen

(<http://de.selfhtml.org/>), natürlich gibt es auch über HTML eine Vielzahl lesenswerter Bücher.³³

Uns geht es hier ja darum, gerade nur das Notwendigste zu vermitteln, das man wissen muss, um Joomla! gut zu verstehen und sinnvolle Informatik dabei zu lernen.

Andernfalls könnte man ja zu jedem der hier besprochenen Themen ein eigenes Lehrbuch lesen, was aber doch sehr aufwändig wäre.

Trotzdem müssen wir jetzt kurz abschweifen, um wieder – im wahrsten Sinne des Wortes - ein wenig in die Tiefe von HTML zu kommen:

Wir verzichten wieder auf die historische Dimension, die in diesem Zusammenhang besonders interessant wäre und erzählen nur die Quintessenz: Es hat sich herausgestellt, dass die weltweite Entwicklergemeinde nicht bereit war einen

³¹ Auch hier empfehle ich vor allem <http://de.selfhtml.org/php/index.htm>

³² Es gibt aber auch handliche kleine Bücher, die man zum Nachschlagen neben dem PC liegen haben könnte: z. B. Cornelia Boenigk und Friedhelm Betz: PHP 5. München 2004. Oder umfassende Werke, die zeigen, wie man mit PHP und MySQL ein CMS bauen kann: Michael Schroer: Web Content Management mit PHP und MySQL. Bonn: 2004.

³³ Besonders erfrischend geschrieben, relativ schlank im Umfang und doch kompetent im Wesentlichen ist: Bruce Lawson und Remy Sharp: HTML 5. München: 2011.

einheitlichen HTML-Code zu akzeptieren. Der strikte Code von XHTML, der HTML 4 ablösen hätte sollen, konnte sich einfach nicht durchsetzen und man musste mit einem durchlässigeren System arbeiten, dem man dann wieder den Namen HTML – diesmal mit der Nummer 5 – gegeben hat.

Ein wesentlicher Unterschied liegt nun darin, dass in HTML 5 den immer wieder eingesetzten Bereichen auf einer Website bestimmte (sprechende Namen) zugeordnet wurden: Diesen Namen hat man danach gewählt, wie schon die DIVs häufig benannt wurden: So wurde der obere Bereich mit der id „header“ benannt: also `div id = header`; in HTML 5 heißt es nun einfach nur `header`. Eine id konnte man früher nur einmal pro Website vergeben, die neuen Begriffe kann man jetzt öfter verwenden und auch ineinander verschachteln. In HTML 4 musste man alle Bereiche als class definieren, die man öfter einsetzen will. Im Prinzip ist es natürlich völlig egal, wie man die Strukturen nennt, die man dann über CSS ansprechen will. Trotzdem ist bestimmt jetzt ein guter Zeitpunkt für eine praktische Übung, um in diesem Begriffsgewirr etwas Klarheit zu gewinnen:

Die SchülerInnen sollten eine ganz einfache Webseite mit dem typischen Muster (Kopf, Navigation, Inhalt und Fußzeile). Die HTML5-Namen dafür sind `header`, `nav`, `article` und `footer` (den folgenden Code sollte man – mit beliebigen Inhalten – auswendig schreiben können):

```
<!doctype html>
<html lang=de>
<head>
<meta charset = utf-8>
<title>Ein guter Titel</title>
<link rel="stylesheet" type="text/css" media="screen"
href="css.css ">
</head>
<body>
<header>
<h1><center>Erste Überschrift</center></h1>
</header>
<nav>
<h2>Menü</h2>
<ul>
<li>Erster Menüpunkt</li>
<li>Zweiter Menüpunkt</li>
</ul>
</nav>
<article>
<h2>Erster Artikel</h2>
<p>Das ist der erste Artikel</p>
</article>
```

```

<article>
<h2>Zweiter Artikel</h2>
<p>Das ist der zweite Artikel</p>
</article>
<footer>
<p><small><center>Das ist die Fußzeile</center></small></p>
</footer>
</body>
</html>

```

Ebenso diesen CSS-Code (selbstverständlich wieder mit beliebigen Eigenschaften):

```

body {
    background-color: gray;
}
nav {
    float: left; width: 20%;
    background-color: aqua;
}
article {
    float: right; width: 79%;
    background-color: orange;
}
footer {
    clear: both;
    background-color: blue;
} 34

```

Jetzt geht es vor allem darum, dass man die Zusammenhänge erkennt. Beginnen wir mit der CSS-Datei (sie ist ja bekanntlich für Formatierung und Layout zuständig). Der CSS-Code besteht aus Regeln, eine Regel besteht aus einem Selektor, über den ausgewählt wird, was von der HTML-Seite formatiert werden soll (z. B. in unserem Fall body) und einem Deklarationsblock, der durch zwei geschwungene Klammern gekennzeichnet ist, in denen die Deklarationen für den jeweiligen Selektor stehen. Diese Deklarationen bestehen aus einer Eigenschaft (z. B. background-color) und einem Wert (z. B. gray), die durch einen Doppelpunkt getrennt sind und von einem Strichpunkt abgeschlossen werden. Wenn man sich das merkt, kann man alle HTML-Strukturen, die man mit dem Selektor auswählt, nach Belieben formatieren. Besonders einfach geht das mit einem CSS-Editor³⁵.

Unser Beispiel sieht sehr hässlich aus. Die SchülerInnen sollten versuchen es zu verschönern und aufschreiben, was sie gerne anders formatieren wollen, aber derzeit noch nicht können, weil sie die dafür notwendigen Befehle noch nicht beherrschen. Möchten sie vielleicht den Innenabstand der Schrift in den Kästen vergrößern, dann

³⁴ Den HTML- und CSS-Code sollten die SchülerInnen auswendig können...

³⁵ Ich verwende den Gratis-Editor TopStyle: <http://topstyle-lite.soft-ware.net/download.asp>

müssten sie sich mit dem Box-Modell vertraut machen.³⁶ Wie man sieht, findet man im Internet sehr rasch Hilfe, selbstverständlich auch, wenn man herausfinden will, wie man den CSS-Code verbessern könnte.³⁷ Aber diese Links habe ich schon kommuniziert. In keinem Buch können alle Lösungen für Webentwickler-Probleme stehen. Im Internet findet man aber tatsächlich (fast) immer Hilfe. Zu lernen im Internet nach Hilfe zu suchen gehört zu den wichtigsten Fähigkeiten eines Webentwicklers. Die SchülerInnen sollten auf das von Anfang an trainiert werden.

Zu diesem Zeitpunkt haben die SchülerInnen – ohne es zu merken – schon eine Menge über HTML und CSS gelernt. In Joomla! werden noch häufig DIV-Container verwendet, die an sich inhaltsleer sind und mit IDs oder Klassen (class) verbunden werden (z. B. `<div id = „header“>` oder `<div class=“article“>`). Der Unterschied zwischen IDs und Klassen liegt nur darin, dass Klassen mehrmals auf einer Webseite vorkommen können, IDs nur einmal. In HTML 5 hat man nun die häufigsten IDs und Klassen einfach als Begriffe eingesetzt, wie man im oben stehenden Code sieht. Auf diese Begriffe kann man genauso wie auf die DIVs über einen Selektor mit CSS zugreifen, um diese zu formatieren. Am besten verwendet man dafür eine eigene CSS-Datei, die man im Head der HTML-Datei einbinden muss, wie wir das auch gemacht haben. All das sollten die SchülerInnen auswendig können, damit sie überall und jederzeit eine HTML-Datei schreiben können und diese mit einer CSS_Datei formatieren können. Wenn man die einfache Struktur unserer HTML-Datei genau durchschaut, fällt es einem nicht mehr so schwer, die index.php des Allrounder-Templates ein wenig zu durchschauen. Und das reicht uns vorläufig auch. Um mehr in die Tiefe gehen zu können, müssen wir uns wieder mit PHP beschäftigen...

2.1.7.Ein Formular mit PHP

Jetzt wird es wirklich schwierig: Was bedeutet es eigentlich, wenn man von einer dynamischen Website spricht? Da kann der Benutzer selbst etwas eintippen und mit dem Eingetippten passiert dann etwas Sinnvolles. Die Dynamik wird durch PHP auf die Website gebracht. Jetzt kommen wir zur „richtigen“ Informatik, zum Programmieren. Zunächst haben wir ein Problem: Wir wollen, dass die Benutzer unserer Website die

³⁶ http://de.selfhtml.org/css/formate/box_modell.htm

³⁷ <http://de.selfhtml.org/css/>

Möglichkeit haben sich über ein Formular anzumelden und die Daten dieser Benutzer sollen dann in einer Datenbank abgespeichert werden, damit wir diese Daten weiter verwenden können. Warum nicht auch in Access, um mit Word einen Serienbrief zu schreiben? Zugegeben, das ist weit hergeholt, weil man wohl immer seltener noch Briefe ausdrucken, in Kuverts stecken und zur Post bringen wird, wenn man die Mitteilungen doch viel schneller und kostengünstiger per E-Mail verschicken kann. Aber vielleicht will man ja über den Serienbrief nur Rechnungen als PDF-Datei ausdrucken und diese dann an eine E-Mail angehängt verschicken, wie auch immer...

2.1.7.1. Zunächst das Formular ohne PHP

Wir haben ja schon darüber geschrieben, dass HTML und PHP sehr eng zusammenhängen. Um es am Anfang nicht gleich zu schwierig zu machen, schreiben wir zunächst nur in HTML den Code für ein einfaches Formular. Auch diesen Code sollten die SchülerInnen auswendig können, da sie dann jedes beliebige Formular aus dem Stehgreif schreiben könnten und die Formularstruktur wirklich genau verstehen würden:

```
<form action=" " method="post">
Vorname: <input type="text" name="Vorname"><br>
Nachname: <input type="text" name="Nachname"><br>
E-Mail-Adresse: <input type="text" name="Email"><br>
Kommentar: <br>
<textarea name="botschaft" cols="50" rows="5">
</textarea><br>
<input type="submit" value="Absenden" name="submit">
</form>
```

Auch zum Thema HTML-Formular kann man sich auf unzähligen Websites im Internet vertiefen. Wir empfehlen immer wieder das gute alte Selfhtml.³⁸

Für unsere Zwecke genügt jetzt aber dieses einfache Formular, da es ja später ohnedies noch viel schwieriger wird.

Im ersten Schritt sollen die SchülerInnen einmal nur versuchen, den vorliegenden HTML-Formular-Code zu verstehen:

Zuerst wird mit dem Formular-Tag <form> das Formular eingeleitet. Abgeschlossen wird es mit dem schließenden Tag </form> In den öffnenden Tag wird aber auch die action = „“ geschrieben (lässt man den Raum zwischen den Anführungszeichen leer, bezieht sich das Formular immer auf sich selbst, schreibt man eine andere Datei in die

³⁸ Siehe <http://de.selfhtml.org/html/formulare/>

Anführungszeichen, passiert die Action dort. Bei der method = "post" werden die Daten versteckt geschickt, bei method = „get“ werden sie an die URL angehängt und somit sichtbar (das wäre bei Passwörtern ungünstig, wenn man die neue URL aber als Lesezeichen verwenden will, wäre „get“ sinnvoll). In den folgenden Zeilen werden die Formularfelder definiert, die in der Datenbank dann den jeweiligen Datenfeldern entsprechen und zuletzt muss das Formular auch abgeschickt werden, dafür ist „submit“ da und value dient nur für die Beschriftung des Abschicken-Buttons. So, das Formular ist also rasch erklärt und man könnte es jetzt beliebig erweitern (z. B. Straße, PLZ, Ort usw.).

2.1.7.2. PHP ist eine Hürde:

Die wahre Schwierigkeit beginnt aber jetzt: Das, was der Benutzer in die Formularfelder einträgt, soll aufgefangen und weiter verarbeitet werden. In meinen bisherigen Unterrichtserfahrungen diverser Praktika war meist hier der Punkt, wo SchülerInnen ausgestiegen sind. An dieser Stelle muss man daher sehr behutsam und langsam vorgehen und bestimmte Begriffe genau erklären, sonst fehlt die Grundlage für alle komplexeren Zusammenhänge rund um PHP. Aus diesem Grund greife ich jetzt auf ein regelrechtes Kinderbuch zurück, in dem die Basics von PHP und SQL sehr anschaulich erklärt werden.³⁹

Die SchülerInnen spüren, dass wir jetzt in den zentralen Bereich der Informatik oder Datenverarbeitung, wie sie ja auch gerne genannt wird, kommen:

```
<?php
if (!empty($_POST['Email'])) {
    $mailbody = $_POST['botschaft'];
    $Email = $_POST['Email'];
    if (@mail("deine_email@mail.at", "Feedback", $mailbody, "From:
$Email")) {
        echo "<p>Danke! Die Botschaft wurde weitergeleitet!</p>\n";
    }
    else {
        echo "<p>Leider gab es einen Sendefehler!</p>\n";
    }
}
?>
```

Es folgt also direkt auf den HTML-Code der PHP-Code. Was bedeutet der aber nun? Die Verzweigungsstruktur if untersucht mit der Funktion !empty, die nicht leer bedeutet

³⁹ Johann-Christian Hanke: PHP und MySQL für Kids. Heidelberg: 2010.

(das Rufzeichen ist wie eine Verneinung), ob der Benutzer in das Formularfeld Email etwas eingetragen hat. Danach wird die eigentliche Botschaft, die der Variablen \$mailbody zugewiesen wird, einem assoziativen Array übergeben (\$_POST['botschaft']), genauso die E-Mail. Mit der Funktion mail wird dann die Botschaft an eine beliebige E-Mail-Adresse geschickt. Der Klammeraffe vor der Funktion mail dient zur Unterdrückung aller Fehlermeldungen, sollte doch etwas nicht funktionieren, wird einfach der else-Zweig ausgegeben.

Zuerst sollte man die Idee der Bedingung mit den SchülerInnen üben, indem man einfach Konditionalsätze bilden lässt und begreifbar macht, was eine Bedingung oder auch mehrere voneinander abhängige Bedingungen bedeuten. Nun ist es auch kein großer Schritt mehr klarzumachen, dass Bedingungen in PHP mit if und else definiert werden.

Danach muss man weitere Fachbegriffe von PHP einführen. So kommen in diesem kurzen Programmtext bereits zwei Funktionen vor. Die Idee der Funktion (das etwas „funktioniert“, indem man den Ablauf der Funktion, der in PHP deklariert ist, einfach abrufen) ist auch noch relativ leicht verständlich. In unserem Fall sind das empty (bzw. das Gegenteil !empty) und mail. Mit einem Transfer auf die Mathematik kann man auf den Begriff der Variablen hinweisen, die bei uns als \$mailbody und \$email definiert sind, und denen jeweils über den Zuweisungsoperator = ein bestimmter Wert übergeben wird. Das ist immer noch recht einfach zu verstehen, für uns aber völlig unbrauchbar, weil wir ja wollen, dass die Daten in eine Datenbank geschrieben werden. Ich denke aber, dass diese einfachen Zusammenhänge wichtig sind, damit die SchülerInnen im Verständnis erste kleine Erfolgserlebnisse haben und auch wichtige Grundlagen für die gleich folgenden komplexen Zusammenhänge aufbauen können.

Wir müssen jedenfalls den Code entsprechend umschreiben, um die Einträge des Benutzers automatisch in eine Datenbank zu bekommen und – ja genau! – wir müssen diese Datenbank auch erst anlegen. Das hatten wir aber schon bei der Joomla!-Installation und noch einmal bei der Joomla!-Sicherung mit Akkeba-Backup. So sieht man, dass dieses Unterrichtskonzept durchaus sinnvoll aufeinander aufbaut.

Wichtig ist, dass die Datenbank genau die Datenfelder definiert hat, die auch als Formularfelder verwendet wurden. Dann benötigen wir den Namen der Datenbank, den Benutzernamen und das Passwort, um uns über die PHP-Datei mit der Datenbank

verbinden zu können und dann müssten die Daten eingegeben werden. Die eingegebenen Daten wollen wir dann gleich in einem HTML-Formular als Tabelle wieder ausgeben.

Hanke verwendet dafür ein geniales Beispiel, nach dem man wesentliche Grundlagen von Joomla! verstanden hat, obwohl es dabei natürlich überhaupt nicht um Joomla! geht, sondern nur um ein schlichtes Eingabeformular, das die eingegebenen Daten in eine Tabelle ausgibt. Beide Dokumente sind über einen Link miteinander verbunden, sodass die SchülerInnen die in dem Formular eingegebenen Daten gleich in der Tabelle nachlesen können.

Hier geht man den entscheidenden Schritt zur Datenbanktabelle, war es doch im vorigen Beispiel nur so, dass die Daten in einer E-Mail an eine zu definierende Adresse geschickt wurden, nur werden sie in eine Datenbank zur weiteren Verarbeitung abgespeichert.

Das Anlegen von Datenbanken sollte kein Problem mehr sein, das Anlegen einer Tabelle ist in phpMyAdmin auch sehr einfach, genauso wie die Definition der einzelnen Datenfelder. Man sollte an dieser Stelle nur das Notwendigste vermitteln, um die Schwelle zu den Datenbanken möglichst niedrig zu halten. Als Felddatentypen sollten für den Anfang VARCHAR, INT und TEXT genügen und die Idee des Primärschlüssels muss natürlich vermittelt werden. Nachdem die SchülerInnen die einzelnen Datenfelder über phpMyAdmin mühsam manuell eingegeben haben, sollte man sie die Tabelle wieder löschen lassen, um eine wesentlich einfachere Methode zu präsentieren: das Kopieren des bereits vorgegebenen SQL-Codes in das entsprechende Fenster in phpMyAdmin:

```
CREATE TABLE adressen (  
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  Vorname VARCHAR(20),  
  Name VARCHAR(20),  
  Str VARCHAR(20),  
  PLZ CHAR(5),  
  Ort VARCHAR(30),  
  Tel VARCHAR(25),  
  EMail VARCHAR(30),  
  WWW VARCHAR(30),  
  Notizen TEXT  
  ) DEFAULT CHARACTER SET utf8;40
```

⁴⁰ Johann-Christian Hanke: PHP und MySQL für Kids. Heidelberg: 2010. S. 289.

Über Einfügen können die SchülerInnen nun schon einige Adressen eintragen und diese über Anzeige in phpMyAdmin anschauen. So wird auch schnell klar, dass jetzt nur noch die Webschnittstellen in Form eines Formulars und einer Tabelle fehlen, um das Ganze sehr komfortabel zu machen.

Dieses einfache Beispiel erklärt aber schon sehr viel Joomla!: Auch Joomla! besteht aus Tabellen in einer Datenbank (sogar über 60!). Auch in Joomla! werden die Inhalte in die Tabellen über Webschnittstellen in die Datenbank eingetragen und ebenfalls über Webschnittstellen wieder für den User schön formatiert ausgegeben. Wir machen hier also eine ideale Übung zum Verständnis der Funktionsweise von Joomla!.

An dieser Stelle wäre es für den Lehrer natürlich verführerisch gleich die SQL-Abfragen mit den SchülerInnen durchzugehen. SELECT, FROM und WHERE sind einfach zu verstehen und auch ORDER BY. Was spricht also dagegen? Ich glaube, dass man sich jetzt auf etwas anderes konzentrieren sollte, um immer das Grundverständnis für das Wesentliche im Auge zu behalten:

Es geht nun vor allem um die Frage, wie man mit PHP auf die Datenbank zugreifen kann und die über HTML eingegebenen Daten in die Datenbank eintragen lassen kann. Wer das verstanden hat, hat viel verstanden, deshalb sollte man jetzt den Fokus darauf legen.

Aufbauend auf den zuletzt durchgenommenen PHP-Code ist der folgende Code in den ersten Zeilen nicht allzu schwer zu verstehen:

```
<?php
$db = @mysqli_connect("localhost", "root", "") or die("Verbindung zu MySQL
gescheitert!");
mysqli_set_charset($db, 'utf8');
@mysqli_select_db($db, "team") or die("Datenbankzugriff gescheitert!");
```

Bis hierhin geht es nur darum, dass man sich mit der Datenbank verbindet und den Zeichensatz aussucht. Das bekommen die SchülerInnen nach meiner bisherigen Erfahrung noch gut hin. Ab jetzt wird es aber schwierig und man muss jede Zeile genau erklären. Wer diesen Abschnitt versteht, hat einige Komponenten der (Datenbank-)Programmierung gut verstanden. Immer wieder werden Werte Variablen übergeben und diese dann weiter verarbeitet. Dieses Prinzip muss man einfach vermitteln:

```
$sql = "SELECT * FROM adressen";
```

Der Variablen `$sql` wird das Ergebnis aus der Auswahl aller Datensätze aus der Tabelle Adressen zugewiesen.

```
$result = mysqli_query($db, $sql);
```

Der Variablen `$result` wird das Ergebnis aus der Abfrage der Datenbank zugewiesen, wobei dieser als Parameter die Variablen `$db` (Verbindung zur Datenbank) und `$sql` (Auswahl aller Datensätze) beigegeben werden.

```
$zeilen = mysqli_num_rows($result);
```

Der Variablen `$zeilen` werden über die Funktion `mysqli_num_rows` die einzelnen Datensätze zugewiesen, die jeweils in eine Tabellenzeile eingetragen werden. Der komplexeste Teil des Codes folgt aber jetzt:

```
// while-Schleife Anfang
while ($row = mysqli_fetch_assoc($result)) {
    echo "<tr>"; // Zeile erzeugen
    // foreach Anfang:
    foreach ($row as $key => $value) {
        echo "<td>$value&nbsp;</td>\n";
    } // foreach Ende
    echo "</tr>"; // Zeile schließen
} // while Ende
mysqli_close($db);
?>
```

Zwei ineinander verschachtelte Schleifen: Mit der Foreach-Schleife werden die einzelnen Zeilen ausgelesen, mit der While-Schleife werden diese dann mit der Funktion `mysqli_fetch_assoc` in assoziative Arrays umgewandelt und zeilenweise in die jeweils richtigen Datenfelder in die Datenbank-Tabelle eingetragen, die mit `mysqli_close` schließlich geschlossen wird.

In allen meinen bisherigen Unterrichtspraktika ist dieses Beispiel gut angekommen, wenn es auch in der relativ kurzen Zeit nicht gelungen ist, den Code in allen Details wirklich zu verinnerlichen, dafür bräuchte man mehr Zeit und einfach auch mehr Wiederholungen. Es gelingt sehr rasch, dass die SchülerInnen Datenbanken anlegen und sich mit diesen Datenbanken verbinden und in PHP Dateien eintragen, die vom Formular in die Datenbank-Tabelle wandern.

Ich glaube, dass die SchülerInnen dieses Beispiel wirklich in allen Details beherrschen sollten, bevor man im Stoff weitergeht, weil es die Grundlage für alle komplexeren Datenbank-Übungen ist.

Wo liegen nun die konkreten Schwierigkeiten der Vermittlung?

Interessanter Weise ist mir bisher aufgefallen, dass der Wissens-Transfer vom Mathematik-Unterricht häufig nicht funktioniert und die SchülerInnen Probleme mit dem Begriff Variable haben. Offensichtlich ist es vielfach immer noch so im Mathematik-Unterricht, dass Beispielabläufe eingeübt werden, tatsächliches mathematisches Verständnis aber weniger trainiert wird, was für das weitere Leben aber wesentlich sinnvoller wäre. So gesehen ist die derzeit laufende Reform des Mathematik-Unterrichts sehr zu begrüßen und offensichtlich auch als Reflex auf ähnliche Erfahrungen wie meine zu sehen. Wir der Begriff der Variablen endlich wirklich verstanden, kommt man zur nächsten Herausforderung, wo es darum geht, bestimmte Werte innerhalb des Programms Variablen zu übergeben, um den Programmcode schlanker zu machen. Sobald diese Technik verinnerlicht wurde kommt man zum schwierigsten Punkt, dem Variablenfeld, also den assoziativen Arrays. Wer die assoziativen Arrays nicht versteht, versteht dynamische Webseiten nicht, sie sind also eine unbedingte Voraussetzung für jede weitere Arbeit in unserem Zusammenhang, da Joomla! ja eine dynamische Website ist. Das Problem liegt darin, dass die SchülerInnen nun zwei Bereiche miteinander verbinden müssen: Auf der einen Seite den Aufbau eines Formulars (deshalb halte ich es auch für sinnvoll, wenn dieser Aufbau auswendig beherrscht wird) und die Technik des assoziativen Arrays, der so heißt wie die Methode, mit der das Formular die eingegebenen Daten verwaltet (z. B. GET oder POST) und als Schlüssel den Namen des Formularfeldes bekommt, dessen einzugebender Inhalt in den Array weiter verarbeitet werden soll und dessen Wert dann eben die Eingabe des Users in das entsprechende Formularfeld ist. Mit diesem Wert wird ja dann weitergearbeitet, da er beispielsweise über PHP in eine Datenbank oder wie in unserem Beispiel in eine Tabelle eingetragen wird (das Datenbanken in Tabellen organisiert werden, ist da nicht mehr viel Unterschied und Hanks Beispiel also wirklich ein genialer Wurf).

Wenn wir auch als Grundlage dieser Unterrichtseinheit ein Buch für Kinder nehmen und wenn wir uns auch viel Zeit mit diesem Stoff nehmen, so wird es doch lange dauern, bis alle SchülerInnen diese Zusammenhänge wirklich verstanden haben. Aus diesem Grund

ist es wichtig, gerade den assoziativen Array regelmäßig zu wiederholen, bei Klassen, die bis zur Matura Informatik haben, vermutlich jedes Jahr. Ich glaube auch, dass es für die LehrerInnen sehr wichtig ist zu erkennen, wo die Schlüsselstellen liegen, die wirklich sitzen müssen, um den Gesamtzusammenhang zu begreifen.

Die nächste Herausforderung ist dann ungleich geringer: Es geht nur noch darum, dass die SchülerInnen bestimmte PHP-Funktionen verstehen und anwenden können. Das kommt eher einem Vokabellernen gleich, wobei es selbstverständlich auch immer im Informatik-Unterricht erlaubt sein muss, dass man nachschaut. Nicht einmal geübte Informatiker können immer alle Funktionen auswendig. Wenn also die Frage aufkommt, wie man Formulareinträge in assoziative Arrays umwandeln kann, kann man die Funktion `mysql_fetch_assoc` leicht finden und anwenden.

Ein größeres Problem stellt in diesen Zusammenhang der Umgang mit Schleifen dar. In unserem Beispiel wurde eine Foreach-Schleife mit einer While-Schleife verschachtelt. Die SchülerInnen müssen in diesem Zusammenhang die Funktionen der einzelnen Schleifen kennen und auch richtig anwenden können. Das bedeutet eine komplexe Verflechtung von Wissen und Kompetenz, die nicht leicht zu erreichen ist, da sie zunächst einmal immer wieder die Funktionen der Schleifen vergessen, wenn sie die Funktionen auswendig können, diese im Kern noch nicht ganz verstanden haben und das ist aber erst die Voraussetzung dafür, dass sie in Zukunft beim Programmieren die richtige Schleife zum jeweiligen Zeitpunkt einsetzen. Die While-Schleife hat eine zentrale Bedeutung im Zusammenhang mit Datenbanken, die den SchülerInnen genau vermittelt werden muss.

An diesem kleinen Beispiel sieht man schon die schwierigen Anforderungen an den Lehrer: Er muss zunächst einmal selbst die Fachkompetenz haben, dann muss er in der Lage sein, diese Fachkompetenz auch zu vermitteln und zuletzt braucht man noch viel Geduld, bis dieses Ziel tatsächlich erreicht ist. Meiner Meinung nach wird auf diese Aspekte auch im Rahmen des Lehramtsstudiums viel zu wenig hingewiesen. Gerade hier liegt aber der Schlüssel für einen erfolgreichen Unterricht, also einen Unterricht, wo die SchülerInnen die vermittelten Lernziele tatsächlich und nicht nur scheinbar erreicht haben. Die wesentlichen Lernziele für das vorliegende Beispiel fasse ich noch einmal kurz zusammen:

- a) Variablen setzen und anwenden können

- b) die Bedeutung von assoziativen Arrays verstehen und diese richtig anwenden können
- c) die Logik von Schleifen verstehen und diese richtig anwenden können
- d) den Zusammenhang zwischen HTML und PHP verstehen
- e) den Zusammenhang zwischen PHP und SQL verstehen
- f) die Struktur einer Datenbank erkennen und Wirklichkeiten in strukturierte Datenbanken umsetzen können

Keine Frage, hier handelt es sich um eine große Herausforderung, der man sich aber stellen muss, wenn man Joomla! in seiner Grundstruktur verstehen will und wenn man Joomla! mit Plugins, Modulen und Komponenten erweitern will, wobei man diese natürlich auch zur Gänze selbst programmieren könnte oder bestehende Elemente in Teilen adaptieren könnte.

Mit diesem Beispiel schließe ich die Beispiele von möglichen Unterrichtseinheiten ab. Leider hatte ich bis jetzt nicht mehr Möglichkeiten mit SchülerInnen zu arbeiten und an dieser Stelle sollten nur bereits erprobte Unterrichtseinheiten stehen.

Nun wende ich mich einem nicht minder komplexen Thema zu, mit dem ich noch keine praktischen Erfahrungen gesammelt habe, das ich aber für besonders wichtig halte.

2.2. Barrierefreie Webseiten im Unterricht:

2.2.1. Zur Theorie:

Dominik Hagelkruys hat in seiner Diplomarbeit „Barrierefreiheit als Thematik im österreichischen Informatikunterricht“⁴¹ eindrucksvoll nachgewiesen, dass das Thema Barrierefreiheit weder im Lehrplan, noch in den Schulbüchern, noch im tatsächlichen Unterricht der LehrerInnen (er hat darüber eine Umfrage gemacht) eine große Rolle spielt. Das wundert mich aus zwei Gründen: Erstens leben wir in einer Zeit, wo gerade

⁴¹ Dominik Hagelkruys: Diplomarbeit „Barrierefreiheit als Thematik im österreichischen Informatikunterricht. Wien: 2012.

auf den Umgang mit Menschen, die besondere Bedürfnisse haben, mehr geachtet wird denn je. Zweitens kann man gerade im Zusammenhang mit der Barrierefreiheit sehr interessante Unterrichtseinheiten machen, wo man das Strukturieren von Inhalten und den Umgang mit semantischem HTML übt. Drittens werden zunehmend Handys als Ausgabegeräte von den Internet-Usern verwendet, auf der anderen Seite aber auch große Bildschirme, die auch als Fernseher genutzt werden. Dies ist eine besondere Herausforderung für barrierefreies Internet.

Hagelkruys schlägt einige sehr interessante Unterrichtseinheiten vor, mit denen man das Thema vermitteln könnte. Angesichts der Tatsache, dass ich in meinem Unterrichtskonzept schon sehr viel Zeit für die Vermittlung von Grundlagen verwenden muss, die als Voraussetzung für die verständnisreiche Arbeit mit Joomla! notwendig sind, muss ich meinen Fokus von der Breite auf die Tiefe legen und schlage folgende Maßnahmen vor:

- a) Die SchülerInnen sollen darauf achten, dass sie nur mit relativen Größen arbeiten, damit die Webseiten sich flexibel an die Bildschirmgröße anpassen, wobei man natürlich eine maximale Größe für die Bild-Elemente festlegen muss, weil man sie nicht zu groß hochladen darf (für sehr große Bildschirme optimiert), wenn die Website mit einer langsamen Handyverbindung auch noch angemessen funktionieren soll. Damit ist aber auch gesichert, dass der User die Schriftgröße selbst größer stellen kann.
- b) Mit spezieller Software sollen die SchülerInnen an der Farbanpassung arbeiten. Gute Hilfen dazu findet man im Internet.⁴² Besonders sinnvoll ist es, die jeweilige Komplementärfarbe zur Hintergrundfarbe als Schriftfarbe zu verwenden, wobei man hierbei eigentlich noch auf verbreitete Farbsehstörungen achten sollte.
- c) Mein Hauptaugenmerk werde ich im Unterricht aber auf das Üben der Strukturierung der Webseite lenken, damit sie mit einem Screenreader gut gelesen werden kann. Der Sreenreader liest auch die Alt-Tags vor, womit klar ist, dass die SchülerInnen Bilder und Videos routinemäßig verbalisieren müssen.

Die in Punkt 3 genannten Anforderungen setzen einen unglaublich großen

⁴² <http://colorschemedesigner.com/>

Zeitaufwand voraus, der im Informatikunterricht alleine überhaupt nicht bewältigbar ist. Wenn man bedenkt, wie verbreitet vielfach schon die Probleme der SchülerInnen in der 5. Klasse im Bereich des sinnerfassenden Lesens ist, kann man sich vorstellen, wie schwierig es für sie ist, einen Text in sinnvolle Einheiten zu gliedern. Die nächste Herausforderung ist es dann, diese Strukturierung in HTML 5 umzusetzen, das heißt zu überlegen, wo man <sections>, Überschriften und <articles> setzt und zuletzt muss man den SchülerInnen noch begreiflich machen, was man unter semantischem HTML versteht und wie man das richtig anwendet.

Ich weise darauf so detailliert hin, weil man oft in didaktischen Handreichungen liest, dass man mit den SchülerInnen den Text sinnvoll strukturieren soll, es steht aber nicht, wie komplex das eigentlich ist und wie schwierig es daher zu vermitteln ist.

2.2.2. Ein praktisches Beispiel:

Ich werde das in der Folge am Beispiel der hier vorliegenden Arbeit von mir bis Seite 40 versuchen. Wie könnte man also diesen Text in einer HTML-Seite inhaltlich strukturiert vorlegen?

```
Vorbemerkung
Konzept
  Vorwort
    Kein Inhalt
    Lehrplan
    Inhalt
      Lernziele
        CMS
        CMS-Struktur
        Template
        Template-Beurteilung
        Smartphones
        PHP
        SQL
        CSS
        JavaScript
        Suchmaschinen-Optimierung
      Unterrichtseinheiten
        Joomla!-Einrichtung
        Joomla!-Struktur
        Joomla!-Blackbox
          Erweiterungen
          Datenbanken
        HTML und CSS
        PHP
```

```

DIV-Container
Formular
    HTML-Teil
    PHP-Teil
        Variablen
        Assoziative Arrays
        Schleifen
        HTML-PHP
        PHP-SQL
        Datenbanken
Barrierefreiheit
    Theorie
        Größen
        Farben
        Strukturen
    Praxis
    Unterrichtsprojekt

```

Wer meint, dass diese einfache Strukturierung, die unseren Abschnittskapiteln entspricht, reicht, täuscht sich leider. Die einzelnen Punkte sollten nun auch noch in Sektoren gegliedert werden. So ergibt sich die Frage, welche Punkte man unter bestimmte Sektoren einreihen könnte. Innerhalb dieser Sektoren müsste man auch Überschriften definieren, die theoretisch von h1 bis h6 reichen könnten. Die nächste Frage wäre, wie man semantisches HTML verwenden könnte, also Struktur-Begriffe, die auch in einem sinnvollen Zusammenhang mit dem Inhalt stehen. Welche neu geschaffenen HTML5-Begriffe könnte man verwenden, wäre es sinnvoll eigene IDs zu definieren, welche Bereiche sollte man in Klassen fassen?

Sobald die Strukturierungen abgeschlossen sind, müsse man sich überlegen, wie man die einzelnen Strukturen über eine externe CSS-Datei layoutiert und formatiert.

Man sieht, dass man sich allein zu diesen Punkten eine Menge überlegen könnte, aber natürlich dabei auch immer zu unterschiedlichen Ergebnissen kommen wird, die man dann einer anspruchsvollen Bewertung unterziehen sollte.

Der nächste Schritt führt uns dann schon vom Block-Style zum Inline-Style. Gibt es Bereiche, die wir mit Span-Tags oder Em-Tags auszeichnen wollen?

Man kann sich leicht vorstellen, dass die SchülerInnen bei dieser Arbeit bald geistig aussteigen werden. So müsste man zunächst die Strukturierung mit sehr einfachen Texten probieren und sich dann zu immer komplexeren Beispielen vorarbeiten.

2.2.3. Strukturierung als Unterrichtsprojekt:

Wie schon gesagt, könnte der Informatik-Lehrer in seinen beiden Wochenstunden das Strukturierungsproblem lange nicht lösen und es würde sehr viel Informatik-Unterricht dadurch verloren gehen.

Wenn die SchülerInnen allerdings den Lernstoff von allen Unterrichtsfächern auf ihrer Joomla!-Website organisieren, könnten sie auch mit anderen FachlehrerInnen an der sinnvollen Strukturierung eben dieses Lernstoffs arbeiten.

Diese Strukturierung beginnt schon damit, dass jedes Unterrichtsfach einer Joomla!-Kategorie zugeordnet wird, aber theoretisch auch noch beliebige Unterkategorien erstellt werden können. Innerhalb einzelner Beiträge könnten ebenfalls Abschnitte strukturiert werden, die Unterüberschriften haben, zu denen man über ein Ankersystem am Anfang des Artikels über einen einfachen Mausklick kommt.

Die Strukturierung soll dann auch mit einem Screenreader getestet werden, damit noch ein eventuelles Feintuning erfolgen kann.

Eine besondere Herausforderung für die SchülerInnen wird in diesem Zusammenhang noch die Beschreibung der Bilder und Videos über die alt-Tags sein. Oft werden diese Beschreibungen ungenügend oder oberflächlich sein. Hier muss den SchülerInnen die moralische Verantwortung klargemacht werden und auch die nötige Fertigkeit, die man für eine gute Beschreibung eines Bildes oder Videos benötigt, muss man genau vermitteln.

Solche Arbeiten können und sollen im Grunde in jedem Fach erfolgen, da jeder Lehrer ein Interesse haben sollte, dass fachspezifische Bilder oder Videos gut beschrieben werden können. Die Hilfe bei der Strukturierung wird durch andere FachlehrerInnen – wenn überhaupt – allerdings größtenteils nur sehr eingeschränkt funktionieren, weil sie die Joomla!-Strukturen nicht kennen werden und auch über semantisches HTML in der Regel weniger als die SchülerInnen wissen werden. Trotzdem müssten die LehrerInnen einiges zur guten Strukturierung einer Webseite beitragen können und sollten mit den SchülerInnen wenigstens etwas darüber diskutieren.

Dieser Abschnitt sollte zeigen, dass allein hinter dieser Aufgabe ein gewaltiger Zeitaufwand steckt und man muss dafür auch die Bereitschaft zur Zusammenarbeit einiger LehrerInnen voraussetzen (ein wenig sollten sie sich vom Informatiklehrer sogar für diese Aufgabe auch coachen lassen), was in der schulischen Praxis gerade in der AHS

noch nicht so die Regel ist, da keine Planungszeit finanziert wird und die meisten LehrerInnen die Vorbereitungszeit für ihre Unterrichtsstunden nur zu Hause halten wollen.

Man sieht also auch, dass viele Unterrichtsvorhaben, die man zunächst einmal leicht vorschlagen kann, in der Praxis gar nicht einfach verwirklicht werden, weil eben die Zeit und die Bereitschaft anderer LehrerInnen zur Zusammenarbeit dafür fehlen, wenn man so weit in die Tiefe gehen will, dass die SchülerInnen tatsächlich auf Dauer eine Kompetenz dabei erwerben.

Da das Thema Barrierefreiheit und Strukturierung aber nicht nur sehr wichtig, sondern auch sehr lehrreich für die SchülerInnen ist, sollte man alles daran setzen, Zeitressourcen und LehrerInnen zur Mitarbeit zu gewinnen, um dieses ambitionierte Unternehmen durchführen zu können.

2.2.4. Barriereelos zur Handyansicht:

Meiner Meinung nach ist es auch sehr sinnvoll, wenn man SchülerInnen beim Unterrichten dort abholt, wo sie gerade stehen. Nun spielt das Smartphone für die meisten SchülerInnen eine große Rolle, also ist es auch sinnvoll dieses im Unterricht einzusetzen. Die derzeit laufende Diskussion über ein Handyverbot an der Schule ist komplett unsinnig, da dieses Verbot kaum exekutierbar wäre und daher leicht umgangen werden könnte und andererseits ist das Smartphone ein Hochleistungsgerät, das für viele Unterrichtsmöglichkeiten eingesetzt werden könnte.

Man darf in diesem Zusammenhang auch nicht übersehen, dass immer mehr über das Smartphone im Internet recherchiert wird und es daher sehr sinnvoll ist, wenn man im Informatikunterricht beim Webdesign auch darauf achtet, dass die erstellten Websites am Handy gut abrufbar sind. Diese Barriere würde sonst bald die Mehrzahl der Internetuser treffen, die nach vorsichtigen Prognosen ab 2015 über mobile Devices surfen wird.⁴³

⁴³ Florence Maurice: Mobile Webseiten. Strategien, Techniken, Dos und Don'ts für Webentwickler. München: 2012. S. XIV.

2.2.4.1. Diskussion der Möglichkeiten:

Auch Experten sind sich derzeit nicht darüber einig, welche Möglichkeit die beste ist: responsive Webdesign, wo die Website erkennt, mit welchem Gerät man zugreift und die dafür beste Ansicht zeigt, mobile WebApps, die aber in der Regel im Internet nicht verlinkt werden und so nicht leicht gefunden werden können oder gar separate Webseiten, die extra für Smartphones erstellt werden.

JavaScript hat in dieser Entwicklung eine unglaubliche Renaissance erlebt, da es nun ja wieder verstärkt darum geht, clientseitig Prozesse zum Laufen zu bringen.

Man darf auch nicht übersehen, dass derzeit der Internetzugang über das Handy, wenn man kein WLAN zur Verfügung hat, noch nicht so schnell ist wie über eine Desktop-Verbindung. Vergleicht man derzeit Smartphones mit Desktop-PCs, so fallen einem zunächst viele Nachteile für den User auf, adaptiert man die Webseiten aber richtig können Mobilität mit GPS und der raschen Möglichkeit zur telefonischen Kontaktaufnahme zu schlagenden Vorteilen werden.

Ich denke, dass man Websites in Zukunft zunächst für mobile Geräte gestalten wird (man wird also auf das Wesentliche fokussieren) und auf dieser Website dann eine Desktop-Version aufbauen. Bei großen Webprojekten ergibt sich das Problem, dass Designer und Webentwickler in starker Wechselseitigkeit voneinander abhängig werden, wobei der Designer durch die Vorgaben des Webentwicklers auch stark beschnitten werden könnte.

Für die derzeitige Unterrichtspraxis ist es wohl am besten, wenn man mit serverseitiger Erkennung arbeitet, wo verschiedene Header ausgewertet werden, die bei der Kommunikation mit http automatisch vom anfordernden Gerät mitgeschickt werden. Es gibt verschiedene Skripte, die helfen zu erkennen, ob der Zugriff über ein mobiles Gerät erfolgt.

Während ich an dieser Arbeit schreibe, kann sich aber gerade in diesem Bereich viel entwickeln, was derzeit noch gar nicht voraussehbar ist. Der Zug wird wahrscheinlich in die Richtung gehen, dass serverseitige Detection und clientseitige Feature Detection miteinander kombiniert werden.⁴⁴ Will man die SchülerInnen in diesem Bereich unterrichten, muss man sich –wie eigentlich in allen Bereichen der Informatik – am Laufenden halten, was für viele LehrerInnen eine kaum bewältigbare Aufgabe sein

⁴⁴ Florence Maurice: Mobile Webseiten. Strategien, Techniken, Dos und Don'ts für Webentwickler. München: 2012. S. 362.

dürfte, zumal sie schon im Rahmen ihrer Ausbildung derzeit über die neuesten Entwicklungen wenig informiert werden, da es auch für die UniversitätslehrerInnen nicht immer einfach ist, bei dieser schnelllebigen Entwicklung ständig am letzten Informationsstand zu sein.

Ich denke aber, dass es gerade für Joomla! sehr bald leicht umsetzbare Lösungen geben wird. Darüber werden wir im Abschnitt zur Template-Gestaltung noch näher eingehen.

2.3. Fazit:

Der bis jetzt vorliegende Stoff wird mir großer Mühe und nur in Zusammenarbeit mit KollegInnen im Laufe der 5. Klasse umsetzbar sein. Falls Informatik bis zur Matura weiter läuft, muss man die Inhalte auch laufend wiederholen und festigen, da alle wichtigen Grundlagen für Joomla! erarbeitet werden und auch die einfache Arbeit mit Joomla! laufend geübt wird.

Neben dieser Wiederholung sollte man sich in den folgenden Jahren vor allem auf zwei Arbeitsbereiche konzentrieren: auf die Gestaltung von Templates für Joomla! (also das äußere Erscheinungsbild der Website) und auf die Programmierung von Erweiterungen für Joomla!. Die beiden Themenkomplexe wären auch für die Gestaltung von vorwissenschaftlichen Arbeiten sehr interessant. Die SchülerInnen müssten zu diesem Zeitpunkt dann schon an eigenen Projekten auch zu Hause für Informatik arbeiten. Im folgenden Abschnitt stelle ich einige Arbeitsmöglichkeiten vor, wobei ich zunächst wieder mit ganz einfachen Vorschlägen beginnen werde.

2.4. Komplexe Arbeitsaufgaben mit und für Joomla!:

Der Lehrer steht nun vor einer schwierigen Aufgabe: Welche Beispiele sollen ausgewählt werden, um den SchülerInnen Joomla! verständlich zu machen? Nahe liegend wäre, einfach Joomla!-Erweiterungen zu analysieren oder sogar selbst zu programmieren. Die Sache ist aber leider etwas komplexer. All diese Aufgaben werden nämlich rasch sehr schwierig und die SchülerInnen könnten die Lust an der Arbeit verlieren. Macht man aber gar nichts in der Richtung, geht der Zug zum Joomla!-Experten verloren, der ja gerade durch die Arbeit mit Joomla! von Anfang an in Gang gesetzt werden soll. Es gibt

aber auch sehr einfache und rasch erfolgreiche Übungen zu PHP, SQL, HTML, CSS und JavaScript (also den wesentlichen Bereichen, aus denen Joomla! besteht), die ebenfalls zum Ziel führen können. Die Lösung dieses Problems liegt – glaube ich – in einer zweigleisigen Vorgangsweise. Man sollte also immer wieder spezifisch nur auf Joomla! hinarbeiten, dazwischen aber auch kleine Arbeiten zu den „Bauteilen“ von Joomla! machen, durch die man ebenfalls ein Verständnis für das Ganze gewinnt.

2.4.1. Die erste eigene Komponente:

Zu Beginn versuchen wir uns aber gleich am schwierigsten, den Komponenten von Joomla! (es gäbe ja auch die kleineren Module oder Plugins), um den SchülerInnen zu zeigen, wo die Reise hingeht. Joomla! ist ja ein modulares System und es werden laufend Erweiterungen für Joomla! programmiert.⁴⁵ Es ist aber völlig klar, dass man die Funktionsweise dieser Erweiterungen und damit den Aufbau von Joomla! überhaupt, besser versteht, wenn man sich selbst an die Programmierung einer Komponente heranwagt.

Die große Herausforderung dabei ist natürlich, dass man von Anfang an algorithmisch denken muss: Welches Problem habe ich, wie kann ich es lösen? Häufig wird es auch schon fertige Erweiterungen geben, die mein Problem lösen (vielleicht aber auch erst nach einer geringfügigen Adaptierung).

Zunächst sollten die SchülerInnen aber nur die Form der Komponenten nachbauen können, damit sie diese dann später mit anspruchsvollen, vielleicht sogar selbst entwickelten, Inhalten füllen können. So beginnen wir mit einem Projekt in Anlehnung an Reibolds Vorschlag einer leeren Komponenten-Hülse.⁴⁶

Als Einstieg sollten die SchülerInnen aber wesentliche Funktionen von phpMyAdmin wiederholen, indem sie eine Probedatenbank anlegen und folgende Arbeitsaufgaben erledigen:

- a) Erzeuge, lösche und erstelle eine Datenbank
- b) Erzeuge, kopiere, lösche bearbeite und benenne eine Tabelle um
- c) Lösche, editiere und füge Tabellenfelder hinzu
- d) Führe beliebige SQL-Statements aus

⁴⁵ Siehe dazu: <http://extensions.joomla.org>

⁴⁶ Vgl. Holger Reibold: phpMyAdmin kompakt. Saarbrücken: 2009. S.11-25. (Früher <http://www.bomots.de>, jetzt: <http://www.brain-media.de>)

Nach diesen kleinen Fingerübungen, die nur eine Wiederholung darstellen, kann man mit dem eigentlichen Projekt beginnen.

Zunächst müssen sich die SchülerInnen einen Überblick von Joomla! verschaffen. Da sie schon ein Jahr damit arbeiten, sollte dies nicht allzu schwer fallen. Es geht jetzt allerdings auch darum, Dinge anzuschauen, die man im täglichen Umgang mit dem CMS nicht benötigt. Wenn man die Unterverzeichnisse Joomla!s anschaut, findet man sowohl im Frontend als auch im Backend das Verzeichnis components, in dem – wenig überraschend – die Dateien für die Komponenten untergebracht werden. Dort gibt es schon nach der Grundinstallation 11 Unterverzeichnisse wie com_banners für die Banner-Verwaltung oder com_contact für die Kontakte. Diese Komponenten sind über das Komponenten-Menü des Backends verfügbar.

Will man eine Komponente erstellen, muss man sieben Schritte befolgen:

- a) Die Komponenten muss für das Frontend und Backend erzeugt werden
- b) Die Komponenten muss in der Datenbank registriert
- c) Es muss ein entsprechender Link im Frontend erzeugt werden
- d) Man muss eine Werkzeugliste herstellen
- e) Man benötigt eine Datenbanktabelle, die man extra erstellen muss
- f) Es müssen Funktionen für die Datenbankmanipulation implementiert werden
- g) Es müssen Funktionen für die Darstellung im Frontend implementiert werden

Es sind also einige Punkte zu bearbeiten. Den SchülerInnen wird bei der Arbeit aber bestimmt rasch klar, dass sie dadurch wirklich wesentliche innere Zusammenhänge von Joomla! verstehen lernen, die sie bis jetzt zwar unbewusst angewandt haben, aber in ihrer tatsächlichen Funktionalität noch nicht verstehen konnten.

Dieses Projekt konnte ich in der Praxis leider noch nicht durchführen, da ich hierfür ein Jahr Vorlaufzeit benötigen würde, um alle notwendigen Grundlagen zu festigen. Im Prinzip wird aber darauf jetzt nur aufgebaut und es kommen einige neue Aufgaben dazu:

Schreiben wir eine Komponente, die lediglich „Erste Komponente“ ausgibt, ist das schon ein gewisser Aufwand. Auf diese Komponenten kann man dann aber komplexere Anwendungen problemlos aufbauen.

Zunächst erstellt man im Ordner Joomla!/components den Ordner com_erste. Mit einem beliebigen Editor wird dann folgende PHP-Datei geschrieben:

```
<?php
defined( '_JEXEC' ) or die (,'Restricted access');
echo ,<div class="componentheading">Erste Komponente!</div>;
?>
```

Im Frontend ist die Komponente bereits über die URL

http://../Joomla/index.php?option=com_erste abrufbar.

Für das Backend muss der Ordner com_erste im Verzeichnis

/Joomla!/administrator/components erstellt werden. Dort wird die Datei

admin_erste.php abgespeichert, die wie folgt aussieht:

```
<?php
defined( '_JEXEC' ) or die (,'Restricted access');
echo ,Erste Komponente!';
?>
```

Nun ist die Komponente auch über das Backend verfügbar und zwar über die Adresse

http://../Joomla/administrator/index.php?option=com_erste

Nun muss eine Integration in die Joomla!interne Navigation hergestellt werden, damit die Komponente über das Backend verwaltet werden kann und über das Frontend aufgerufen werden kann. Zu diesem Zweck muss die Komponente in der Datenbank registriert werden. Dafür gibt es zwei Wege. Nun sind wir schon wieder im Bereich der Wiederholung des Abschnitts Datenbank-Arbeit mit SQL bzw. mit phpMyAdmin:

Entweder gibt man im SQL-Fenster folgenden Code ein (vorausgesetzt das Datenbank-Präfix ist jos_:

```
INSERT INTO jos_components (name, link, admin_menu_link,
admin_menu_alt, ,option', admin_menu_img, params)
VALUES
(,Erste Komponente', ,option=com_hello', ,Manage Erste', ,com_hello',
,js/ThemeOffice/component.png', ,');
```

Natürlich kann man alternativ auch in phpMyAdmin die Datenbank von Joomla! öffnen und dort die Tabelle jos_components aufrufen. Über Einfügen füllt man das zugehörige Formular komfortabel mit den Daten wie oben aus. Hat man den Code allerdings schon

vorliegen, geht es natürlich schneller, wenn man dieses über das SQL-Fenster einfach hineinkopiert (siehe unser entsprechendes Beispiel über die Formulare).

Über die Joomla! -Administration ist die Komponente nun schon über das Components-Menü aufrufbar.

Nun muss die Komponente im Frontend eingebunden werden; das ist eine einfache Arbeit, die den SchülerInnen schon bekannt sein muss, da sie ja schon seit einem Jahr mit Joomla! arbeiten und bereits fertig programmierte Komponenten heruntergeladen haben und in das Menü eingebunden haben.

Nun wird die Komponente noch mit den typischen Funktionsbuttons ausgestattet, die das Sichern, Löschen, Editieren und Veröffentlichen erlauben. Dafür wird die Datei `toolbar.erste.html.php` im Ordner `/Joomla/administrator/components/com_erste` kopiert (den Code stelle ich im Anhang und zum Download zur Verfügung).

Mit weiterem Code muss auch bestimmt werden, was dargestellt werden soll. Auch den Inhalt dieser Datei `toolbar.erste.php` stelle ich im Anhang und zum Download zur Verfügung.

Nun muss noch die Datenbanktabelle erstellt werden, in der die Daten gespeichert werden:

```
CREATE TABLE ,jos_erste' (  
  ,id' INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY, ,message' TEXT NOT NULL ,  
  ,published' TINYINT(1) NOT NULL  
);
```

Zuletzt muss noch eine Tabellenklasse erzeugt werden. Dabei wird die vordefinierte Klasse `JTable` verwendet, die die Funktionen für die Datenmanipulation einzelner Tabellen in der Datenbank bereitstellt. Mit diesem Schritt wird die Klassenfunktionalität einfach erweitert. Man erzeugt einen Ordner `tables` innerhalb von `/Joomla/administrator/components/com_erste`, wo die Datei `erste.php` abgespeichert wird (den Code dafür und für einige Funktionalitäten stelle ich wieder im Anhang und im Download zur Verfügung).

Nun kann man beliebige Funktionen der Komponente zuschreiben.

Ich würde den Code für die SchülerInnen zur Verfügung stellen, sodass sie das Ganze zunächst nur zusammenkopieren und richtig abspeichern müssen. Wenn sie die Basis gut verstanden haben, können sie versuchen, neue Funktionalitäten der Komponente

zuzuschreiben oder nach dem vorgegebenen Muster komplett neue Komponenten programmieren.

In der Praxis wird das selten nötig sein, das es unzählige fertige Komponenten für alle nur denkbaren Anwendungen zum Gratisdownload bereits gibt.⁴⁷

Man sieht, dass dieses Beispiel zwar nicht ganz einfach ist, für routinierte Joomla!-Administratoren aber in einer Doppelstunde zu schaffen sein müsste, wenn sie den Code zur Verfügung gestellt bekommen.

Im Code ist schon teilweise JavaScript enthalten, das vorläufig noch als Blackbox genommen werden kann. Die SchülerInnen werden dadurch schon auf diese Programmiersprache neugierig gemacht. In Kürze werden sie mehr darüber erfahren, was ihnen auch bei eigenen Programmierprojekten für Joomla!-Komponenten helfen wird. In der Konzeptphase geht es ohnedies einmal darum zu überlegen, was die Komponente können soll, ob es schon ähnliche Komponenten gibt, die man adaptieren könnte oder ob man die Komponente komplett neu programmieren muss. Falls ja, müssen ohnedies zuerst die Algorithmen formuliert werden.

2.4.2. Das Template:

Mit den Templates von Joomla! kann man die SchülerInnen begeistern. So einfach ist es, das Aussehen einer Website zu verändern, deren Inhalt separat über eine Datenbank verwaltet wird. Genauso wie bei allen anderen Joomla!-Erweiterungen kann man diese von unzähligen Seite aus dem Internet einfach herunterladen und installieren oder man kann sie herunterladen, nach eigenen Vorstellungen adaptieren und dann installieren oder man kann auch selbst völlig neue Templates erstellen.

Im ersten Fall bleiben wir bei der reinen Klick-Informatik, und wir sehen die Erweiterungen als Blackbox an, im zweiten Fall benötigt man bei bestimmten Templates kaum Vorkenntnisse, in anderen Fällen schon einige Kenntnisse in HTML, CSS und PHP und im dritten Fall kommen JavaScript, SQL und die Arbeit mit Datenbanken noch dazu.

Templates einfach installieren:

Nichts leichter als das: Man googelt im Internet nach Joomla!-Templates, ladet sie auf den Computer und dann über das Backend über Erweiterungen auf den Server, wo sich

⁴⁷ Vgl. z. B.: <http://www.Joomlaos.de>

Joomla! befindet und stellt unter Erweiterungen-Templates das gewünschte Template auf Default.

Diese scheinbar anspruchslose Tätigkeit kann rasch anspruchsvoll werden, wenn man mit den SchülerInnen die Vor- und Nachteile einzelner Templates bespricht, Module, die noch deaktiviert sind, aktiviert, neue Module in das Template einbaut und mit sinnvollem Inhalt füllt usw.

Man kann also schon auf diesem Bereich zu durchaus anspruchsvollen Ergebnissen kommen. Nach meiner bisherigen Unterrichtserfahrung kommt diese Arbeit auch sehr gut bei den SchülerInnen an und sie sind sogar teilweise gerne bereit zu Hause mit den Templates weiter zu experimentieren.

Natürlich gibt es auch ein paar Fußangeln, auf die man achten sollte. In jedem Fall muss an die Website vor dem Experimentieren mit Templates über Akeeba sichern, da es immer wieder Templates gibt, die die Website komplett zerstören. Noch besser wäre es, über Akeeba eine Kopie von der Website anzulegen, mit der man dann nach Herzenslust experimentieren kann.

Es gibt auch Templates, bei denen man über das Backend mit intuitiven Klicks eine Menge umstellen kann ohne eine Zeile CSS oder HTML gelesen zu haben. Sehr empfehlenswert ist dafür zum Beispiel das Template Allrounder⁴⁸, bei dem man neben vielen Grundeinstellungen auch das komplette Farbensetting neu einstellen kann, indem man die entsprechenden Hex-Codes verändert. Eine ideale Möglichkeit ergibt sich hier mit dem Farbenrad⁴⁹. So können die SchülerInnen nach neuen Farbzusammenstellungen suchen und diese dann mit Allrounder umsetzen. Diese sehr einfache Variante ist unbedingt schon für die 5. Klasse zu empfehlen, weil die SchülerInnen dadurch Joomla! so richtig schätzen lernen. Natürlich gibt es auch andere Templates, wo man über Erweiterte Optionen einiges verstellen kann, es gibt aber wenige Beispiele, wo diese Möglichkeiten so umfassend sind. In der XML-Datei des Templates kann nämlich festgelegt werden, ob und welche Auswahlfelder zur Verfügung stehen sollen. Die meisten Template-Architekten sind natürlich nicht sehr interessiert daran, dass ihr Template bis zur völligen Unkenntlichkeit sehr einfach verändert werden kann. Aus diesem Grund sind diese Möglichkeiten häufig sehr eingeschränkt.

⁴⁸ <http://www.lernvid.com/downloads/allrounder-joomla-16-template.html>

⁴⁹ <http://colorschemedesigner.com/>

Ein wichtiger Punkt bei der Analyse einfach downloadbarer Templates ist natürlich auch die Barrierefreiheit. Mit dieser Analyse könnte man sehr anspruchsvolle Unterrichtseinheiten gestalten und auch auf diesem Weg dieses so wichtige Thema behandeln. Anders als bei den weiter oben genannten Strukturierungsaufgaben kann dies allerdings nur im Informatikunterricht passieren, da der Lehrer einige Vorkenntnisse zum Thema benötigt.

Templates verändern:

Allrounder ist auch ein guter Kandidat, wenn man Templates über CSS, HTML und PHP verändern will.

Es ist aber auch sehr hilfreich, wenn man dafür Templates heranzieht, die in Joomla! bereits vorinstalliert sind: Beez_20 kann man an dieser Stelle schon vergessen, da es noch für HTML 4 optimiert ist, aber Beez_5 wäre ein guter Kandidat, besonders auch deshalb, weil es sich um ein bewusst barrierearm gestaltetes Template handelt. Für den Beginn ist allerdings das Atomic-Template zu empfehlen, da es äußerst sparsam ausgestattet ist und daher eine gute Übersicht über die wesentlichen Bereiche eines Templates ermöglicht. Naturgemäß enthält es auch wesentlich weniger Dateien als das Beez_5-Template, wodurch in weiterer Folge auch der Vergleich der beiden Templates Sinn macht, bzw. auch der Versuch, das Atomic-Template um einige Funktionen zu erweitern.⁵⁰

Um die Sache nicht unnötig zu verkomplizieren, werden wir uns hier nur mit den Frontend-Templates beschäftigen, da es doch in der Regel wenig Sinn machen wird, allzu viel Zeit für die Gestaltung von Backend-Templates zu verwenden, die man ja nur selbst benützt. Natürlich kann dies rasch zum Thema werden, wenn man ein Backend für unterschiedliche Benutzer sinnvoll adaptieren will oder für die Nutzung am Smartphone.

Die Verzeichnisse der unterschiedlichen Frontend-Templates werden alle im Ordner /joomla/templates abgelegt. Schauen wir uns einmal die Verzeichnisstruktur des sehr einfachen Atomic-Templates an: alle Unterordner liegen im Verzeichnis /atomic. Im Verzeichnis /css liegen wenig überraschend die CSS-Dateien des Templates. Alle Angaben zum Grundlayout findet man in der Datei template.css. Das /html-Verzeichnis

⁵⁰ Kerstin Dankwerth und Dana Boldt: Joomla! 2.5 Templates. Mit umfangreichem Praxisteil. Heidelberg: 2012. S. 76.

ist für die Dateien vorgesehen, mit denen man die Core-Ausgabe der Module und Inhalte überschreiben kann (es geht also um die sogenannten Template-Overrides, mit denen man sich nicht gleich am Anfang beschäftigen muss, weil das Thema doch etwas komplex ist und es nicht zwingend nötig ist, die Standardausgabe der Inhalte und Module zu überschreiben, ohne dabei Änderungen an den Joomla!-Core-Dateien vorzunehmen).

Im Verzeichnis /images werden die Bilder für das Template abgespeichert.

Damit ist die grundsätzliche Struktur eines Templates schon beschrieben.

Nun zu den Dateien:

Die index.php ist die Zentrale des Templates; hier müssen alle Erweiterungen eingebunden werden und es handelt sich auch um die Datei, über die die Darstellung der Inhalte aus der Datenbank gesteuert wird.⁵¹ Da es sich um eine PHP-Datei handelt, bietet sie eine ideale Möglichkeit wesentliche Inhalte zu PHP zu wiederholen. Gute PHP-Kenntnisse sind auch nötig, um den Aufbau dieser Datei analysieren zu können.

Die templateDetails.xml ist in dem strikten XML geschrieben und beschreibt den Aufbau des Templates. Bei der Installation im Backend legt der Template-Installer die Verzeichnisse für das Template an und kopiert die Dateien in die entsprechenden Verzeichnisse. Genauso wird die Position der Module in dieser Datei festgelegt.

Die template.css steuert das Aussehen des Templates und legt auch die Formatierungen der CSS-Klassen von Joomla! fest. Hier müssten die SchülerInnen ansetzen, wenn sie Adaptierungen am Layout und der Formatierung vornehmen wollen, die nicht über Klickeingaben vom Template-Designer freigegeben wurden.

Die Datei template_thumbnail.png ist das Vorschaubild des Templates, das im Backend angezeigt wird.

Die Datei template_preview.png ist das große Vorschaubild, das angezeigt wird, wenn man in der Template-Vorschau auf das kleine Bild klickt.

Die Herausforderung für den Lehrer liegt darin, die wesentlichen Dateien in ihrer Funktionalität den SchülerInnen zu erklären. Beginnen sollte man mit der templateDetails.xml, fortsetzen mit der sehr komplexen index.php, um mit der template.css abzuschließen. Ich werde in der Folge versuchen, möglichst einfache, dabei

⁵¹ Vgl. Angie Radtke: Joomla! Templates entwickeln. Barrierefreie & attraktive Designs von Konzept bis Umsetzung. München: 2011. S.127-135.

aber doch gründliche Erklärungsmuster vorzulegen, da wir jetzt zur zentralen Aufgabe der Website-Gestaltung kommen. Wer das schafft, kann eine Joomla!-Website in jeder beliebigen Erscheinung gestalten und die reiche Funktionalität der Erweiterungen nützen und auf alle Inhalte der Datenbank nach Belieben zugreifen: die SchülerInnen werden zu WebdesignerInnen.

Erklärung der templateDetails.xml:

Wenn man sich diese Datei im Atomic-Template anschaut, ist sie fast selbsterklärend:⁵²

Nehmen wir die ersten paar Zeilen:

```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE install SYSTEM "http://www.joomla.org/xml/dtd/1.6/template-install.dtd"
  PUBLIC "-//Joomla! 1.6//DTD template 1.0//EN">
  <-extension client="site" type="template"
  version="2.5"><name>atomic</name><creationDate>10/10/09
  </creationDate><author>Ron
  Severdia</author><authorEmail>contact@kontentdesign.com</authorEmail><authorUrl
  >http://www.kontentdesign.com</authorUrl><copyright>Copyright (C) 2005 - 2013
  Open Source Matters, Inc. All rights reserved.</copyright><license>GNU General Public
  License version 2 or later; see
  LICENSE.txt</license><version>2.5.0</version><description>TPL_ATOMIC_XML_DESCRI
  PTION</description>
```

Die SchülerInnen sollten nun schon in der Lage sein, den vorliegenden Code in Ansätzen zu erklären: Es handelt sich um die Computersprache XML, gekennzeichnet durch den Eingangstag <?xml (...)

Im Starttag wird der Zeichensatz UTF-8 festgelegt und es erfolgt die Angabe des DocType. XML (die striktere Beschreibungssprache als HTML) ist für Joomla! sehr wichtig (sie wird auch für Komponenten, Module, Plugins und Sprachdateien genutzt). Anders als in HTML kann man in XML die Namen für die Tags frei wählen. Die Joomla!-Entwickler haben das auch getan. Will man aber ein eigenes Template herstellen, das mit Joomla! auch tatsächlich kompatibel ist, muss man sich an den Vorgaben der Joomla!-Entwickler orientieren.

Es ist nun bestimmt nicht notwendig gleich alle „Details“ der templateDetails.xml mit den SchülerInnen genau zu besprechen, sondern es ist der bessere Weg das zu klären, was offensichtlich ist: Fragt man also die SchülerInnen, wo Angaben über Name des Templates, Erstellungsdatum, Name und E-Mail des Autors, Webadresse, Copyright,

⁵² Daniel Koch: Das große Buch zu Joomla! 2.5. Vom schnellen Einstieg zum professionellen Webauftritt. Düsseldorf: 2012. S. 559.

Lizenz, Version und Beschreibung zu finden sind, wird das eine leichte Aufgabe sein, wenn man den oben stehenden Code genau anschaut.

Es ist auch die Struktur der Datei gut zu erkennen, sind doch alle Tags unterhalb des DocType im Extension-Tag verpackt.⁵³

Das Gleiche gilt für den folgenden Code:

```
<files><folder>html</folder><folder>css</folder><folder>images</folder><fo  
lder>language</folder><folder>js</folder><filename>index.php</filename><f  
ilename>index.html</filename><filename>favicon.ico</filename><filename>t  
emplateDetails.xml</filename><filename>template_preview.png</filename>  
<filename>template_thumbnail.png</filename><filename>component.php</  
filename><filename>error.php</filename></files>
```

Hier werden in den Tags `<files>` `</files>` alle für das Template relevanten Ordner (folder) und Dateien (filename) aufgelistet.

In diesem Zusammenhang muss den SchülerInnen wieder klargemacht werden, wie wichtig es ist, Ordnung zu halten. Jede Datei, die für das Template benötigt wird, muss auch in der `templateDetails.xml` eingetragen werden. Vergisst man auf eine, gibt es eine Fehlermeldung.

Auch der folgende Code ist leicht verständlich:

```
positions><position>atomic-bottomleft</position><position>atomic-  
bottommiddle</position><position>atomic-  
search</position><position>atomic-sidebar</position><position>atomic-  
topmenu</position><position>atomic-topquote</position></positions>
```

Jedem wird hier rasch bewusst werden, dass es um die Positionierung der einzelnen Elemente, also um das Layout geht. In den Tags `<positions>` `</positions>` werden dieselben definiert.

Vermutlich werden die SchülerInnen auch erraten können, worum es in den Tags `<languages>` `</languages>` geht...

Die Theorie ist also schnell durchschaut. Um die Erkenntnisse zu festigen, sollten gleich praktische Beispiele ausprobiert werden, indem man neue Dateien für das Template erstellt und diese dann richtig in die `templateDetails.xml` einbindet.

⁵³ Vgl. David Jardin: Joomla! 2.5. Professionelle Webentwicklung. München: 2012. S. 212.

Interessant wäre hier die Erstellung eigener Parameter in den Tags `<params>` `</params>`, wo man unterschiedliche Farbvarianten eines Templates generieren könnte. Diese Übung ist sehr einfach und zeigt das Zusammenspiel von XML, PHP und CSS in einer sehr kompakten Art und Weise. Mehr dazu schreibe ich am Ende dieses Abschnittes.

Da diese Datei zwar sehr wichtig, aber eigentlich nur eine Art Inhaltsverzeichnis ist, wird sie von den SchülerInnen leicht verstanden. Wesentlich komplexer ist die folgende Datei...

Erklärung der index.php:

Auf Seite 30 habe ich mit dem Formular-Beispiel schon in den PHP-Code eingeführt. Auch mit diesen Vorkenntnissen wirkt die index.php auf den ersten Blick für Joomla!-Neulinge monströs und – sie ist es auch. Sie stellt die wesentliche Herausforderung dar, Joomla! etwas besser zu durchschauen.⁵⁴ Früher oder später wird man sich dieser Herausforderung stellen müssen, wenn man mit Joomla! arbeiten will.

Für Neulinge ist schwer zu fassen, dass die index.php bestimmte, schon fertig programmierte Teile von Joomla! durch spezielle Befehle in die Website einbindet. So bleibt ein bestimmter Bereich Blackbox und kann zunächst auch nur in der vorgegebenen Weise verwendet werden. Joomla! will seinen Usern aber trotzdem maximale Flexibilität gewähren und bietet die Möglichkeit der Overrides an. Das wird aber nun wirklich kompliziert und soll erst später behandelt werden. Zunächst versuchen wir die index.php so einfach wie möglich zu erklären.

Jede PHP-Datei wird in Joomla! mit folgendem Eingang geschützt:

```
<?php
//no direct access
Defined( „JEXEC“ ) or die;
```

Die Konstante JEXEC wird immer gesetzt, wenn eine Joomla!-Seite angesurft wird. Versucht jemand direkt auf eine PHP-Datei eines Joomla!-Systems zuzugreifen, wird das durch den oben stehenden Code unterbunden.

⁵⁴ Tim Schürmann: Praxiswissen Joomla! 2.5.- Köln: 2012. S. 553.

Wie jede HTML-Seite besteht auch die index.php aus einem <head> und einem <body>, vor dem <head> findet man die üblichen Definitionen.

Im <head> ist es besonders wichtig, die eingebundenen CSS-Dateien zu beachten. Im Atomic-Template werden alle Code-Zeilen vorbildlich kommentiert (in der Folge zur besseren Übersicht fett ausgezeichnet):

```
<head>

    <!-- The following JDOC Head tag loads all the header and meta information
    from your site config and content. -->
    <jdoc:include type="head" />

    <!-- The following five lines load the Blueprint CSS Framework
    (http://blueprintcss.org). If you don't want to use this framework, delete
    these lines. -->
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo
    $this->template ?>/css/blueprint/screen.css" type="text/css" media="screen,
    projection" />
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo
    $this->template ?>/css/blueprint/print.css" type="text/css" media="print" />
    <!--[if lt IE 8]><link rel="stylesheet" href="blueprint/ie.css"
    type="text/css" media="screen, projection"><![endif]-->
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo
    $this->template ?>/css/blueprint/plugins/fancy-type/screen.css"
    type="text/css" media="screen, projection" />
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo
    $this->template ?>/css/blueprint/plugins/joomla-nav/screen.css"
    type="text/css" media="screen" />

    <!-- The following line loads the template CSS file located in the template
    folder. -->
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo
    $this->template ?>/css/template.css" type="text/css" />

    <!-- The following four lines load the Blueprint CSS Framework and the
    template CSS file for right-to-left languages. If you don't want to use these,
    delete these lines. -->
    <?php if($this->direction == 'rtl') : ?>
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo
    $this->template ?>/css/blueprint/plugins/rtl/screen.css" type="text/css" />
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo
    $this->template ?>/css/template_rtl.css" type="text/css" />
    <?php endif; ?>
```

```

        <!-- The following line loads the template JavaScript file located in the
        template folder. It's blank by default. -->
        <script type="text/javascript" src="<?php echo $this->baseurl
        ?>/templates/<?php echo $this->template ?>/js/template.js"></script>
    </head>

```

Die SchülerInnen sollten die oben stehenden Code-Zeilen mit Hilfe der Kommentare schon einigermaßen durchschauen. Über kleinere Details würde ich bei der ersten Konfrontation hinwegsehen, um die Sache nicht von Beginn an zu kompliziert zu gestalten.

Mit der Anweisung `<jdoc:include type="head" />`

wird interner Joomla!-Code in die Website geladen, nämlich der gesamte head der index.php. Das klingt sehr abstrakt und sollte gleich mit weiteren praktischen Beispielen geübt werden:

Warum stellt man beispielsweise nicht die Sprache einfach mit de-de ein, warum wird stattdessen folgender PHP-Code eingefügt?

```
<?php echo $this->language; ?>
```

Im Backend von Joomla! kann man über Erweiterungen die gewünschte Sprache einstellen; aus diesem Grund muss diese Einstellung in der index.php dynamisch sein.

Mit der Syntax jdoc können beliebige Module in die index.php geladen werden.

Untersucht man die index.php darauf, findet man weitere Beispiele:

`<jdoc:include type="modules" name="position" />` lautet der allgemeine Code. Über die Modulposition wird gesteuert, welches Modul wo im Template eingefügt werden soll. Die Modulposition wird dem Modul über das Backend ganz einfach zugewiesen (Erweiterungen – Module – gewünschtes Modul anklicken). Will man also das Suchen-Modul einfügen gibt man folgerichtig ein:

```
<jdoc:include type="modules" name="suche" />
```

Die Menüs werden gleich wie die Module behandelt. Will man beispielsweise das Servicemenü zum Anzeigen der Sitemap an einer Stelle einfügen, gibt man folgenden Code in der index.php ein:

```
<jdoc:include type="modules" name="servicemenu" />
```

Will man Inhalt einbauen, gibt man `<jdoc:include type="component" />` ein.

Es ist aber nicht nur ein Zusammenhang zwischen PHP und Joomla! herzustellen, sondern die Sache ist noch etwas komplexer:

Der Inhalt wird in folgende div eingebaut:

```
<div class="container">
```

Das setzt nun aber wieder HTML- und CSS-Kenntnisse voraus, wird hier doch ein div-Container definiert, dem eine Klasse zugeordnet wird, die wiederum in CSS formatiert werden kann, wenn man über .container auf sie zugreift. Diese Kenntnisse müssen schon im Vorfeld gefestigt werden, da die SchülerInnen sonst überhaupt nur „Bahnhof“ verstehen würden. Auch mit diesen Kenntnissen sind die Zusammenhänge noch komplex genug. Ich denke auch, dass es nicht reicht, diese Zusammenhänge zu erklären, sondern man muss gleich konkrete Beispiele üben, sonst spiegelt sich nichts davon in den Kompetenzen der SchülerInnen.

Bevor man allerdings mit dem Üben beginne kann, muss man noch die CSS-Dateien besprechen, vor allem natürlich die template.css.

Erklärung der template.css:

Diese Datei ist wiederum relativ einfach zu verstehen, daher zeichne ich die sehr einleuchtenden Kommentare einfach einmal fett aus, um die Übersicht zu verbessern:

```
/* Text styles */
.blog-featured h2 {
    font-size: 1.5em;
    margin-bottom: 0em;
}
p.readmore {
    text-indent: 0;
    font-size: .9em;
}
.joomla-footer {
    font-size: .9em;
    margin-bottom: 30px;
}
/* Article functions */
ul.actions {
    clear: both;
    margin-top: -50px;
    float: right;
}
```

```

ul.actions li {
    list-style-type: none;
    float:right;
    margin-left: 10px;
}

/* Login styles */
p#form-login-username label,
p#form-login-password label {
    width: 160px;
    display:block;
}
p#form-login-remember label {
    font-size: .9em;
    font-weight: normal;
    line-height: 25px;
}
p#form-login-remember input {
    float:left;
    margin-right: 5px;
}
form#form-login ul {
    margin: 0;
    padding: 0;
}
form#form-login ul li {
    list-style-type: none;
    margin-left: 20px;
    font-size: .9em;
}

/* Image float styles */
.img-fulltext-left {
    float:left;
    margin-right: 20px;
    margin-bottom: 20px;
}
.img-intro-left {
    float: left;
    margin-right: 10px;
    margin-bottom: 10px;
}

```

```

}
.img-fulltext-right {
    float: right;
    margin-left: 20px;
    margin-bottom: 20px;
}
.img-intro-right {
    float: right;
    margin-left: 10px;
    margin-bottom: 10px;
}

```

Die SchülerInnen sollten nach den Vorkenntnissen, die sie zu diesem Zeitpunkt bereits haben, nicht nur den vorliegenden Code interpretieren können, sondern diesen auch nach Belieben verändern und erweitern können. Man könnte dazu einige Beispiele vorgeben, z. B. verändere die Breite des Login Styles usw.

Die wichtigste Aufgabe liegt darin, die kombinierten Selektoren zu erklären und auch hier wieder mit neu gestellten Aufgaben andere Kombinationen abzufragen, um zu sichern, dass die SchülerInnen nun über die Kompetenz verfügen, beliebige Selektoren miteinander zu verbinden.

2.4.3. Das Joomla-Framework

2.4.3.1. Was versteht man unter dem Framework?

Das Framework ist die magische Blackbox, auf die man sich bei diversen Joomla!-Anwendungen immer wieder bezieht. Natürlich sollte auch der Aufbau dieses Frameworks den SchülerInnen verständlich vermittelt werden.⁵⁵

Dafür sollte man davor schon etwas in PHP programmiert haben, damit die Begriffe Klassen und Methoden klar sind und es nicht passieren kann, dass diese Klassen mit den CSS-Klassen verwechselt werden.

Die Joomla!-Entwickler haben jedenfalls netter Weise eine Reihe von häufig benötigten Klassen und Methoden in dieses Framework implementiert und man muss nun lediglich darauf zugreifen um diese zu verwenden. So gibt es für die Mail-Funktion bereits die

⁵⁵ Eine sehr gute Anleitung bietet z. B.: David Jardin: Joomla! 2.5. Professionelle Webentwicklung. München: 2012. S. 309-364.

Klasse JMAil und man muss nicht auf die PHP-Funktion mail() zurückgreifen. So kann man gleich automatisch auf die Mailzugangsdaten der Joomla!-Konfiguration zurückgreifen.

Zum tieferen Verständnis des Frameworks muss man zunächst das MVC-Pattern verinnerlichen: M steht für Model, V für View und C für Controller. Der Controller beauftragt das Model mit der Manipulation der Daten, View fragt beim Model die Daten an, diese werden dann auch von Model an View übergeben und der Controller beauftragt View mit der Ausgabe der Daten. Alle Erweiterungen des Joomla!-Frameworks basieren auf dieser Grundstruktur.

Es ist kein Fehler, wenn man den SchülerInnen einen kleinen Überblick über die bestehenden Klassen gibt, wobei hier Vollständigkeit natürlich nicht angestrebt werden kann.⁵⁶

Die Klasse JDatabase übernimmt beispielsweise die direkte Kommunikation mit der verwendeten Datenbank. Die Klasse ersetzt auch die Platzhalter in unseren Datenbankabfragen durch das Datenbank-Präfix der jeweiligen Joomla!-Installation. Ich habe schon mehrfach darauf hingewiesen, dass Joomla! ein ausgezeichnetes Medium ist, um mit der Arbeit mit Datenbanken vertrauter zu werden. Dies ist eine gute Gelegenheit, über SQL gleich auch Access-Funktionen zu üben. Wieken⁵⁷ gibt in seinem Buch immer wieder Hinweise, wie SQL-Prozeduren u.a. in Access und in MySQL funktionieren.

Die Klasse JDatabaseQuery übernimmt die Generierung der Datenbankabfrage, JInput dient zur Abfrage der Werte, JDocument ist wenig überraschend für die Herstellung eines Dokuments zuständig und JFile bzw. JFolder dient dem Zugriff auf Dateien oder Ordner.

JForm ist für die Validierung eines Formulars usw.

Ich denke, dass man nicht zu viele Klassen auf einmal vorstellen sollte, aber einige mit dem Hintergedanken, solche auszuwählen, die man dann für ein Programmierprojekt gleich einsetzen kann.

⁵⁶ Siehe dazu in der Joomla! API-Dokumentation: <http://api.joomla.org>

⁵⁷ John-Harry Wieken: erfolgreich SQL lernen. München: 2012. Z. B. S. 48ff.

2.4.3.2. Programmiere eine Komponente für eine Stellenanzeige:

Es ist durchaus sinnvoll eine solche Komponente zu programmieren und vor allem schon deutlich anspruchsvoller als unser erstes Beispiel (vgl. Seite 23).

Im Frontend soll je ein einzelnes Stellenangebot im Menü verlinkt werden, im Backend soll es über die Listenansicht gepflegt werden.⁵⁸

Das Geniale an diesem Projekt ist, dass man nach der Template-Gestaltung wieder ganz ähnliche Mechanismen anwenden kann, die SchülerInnen also schon eine gewisse Routine bekommen. So muss eine bestimmte Verzeichnisstruktur angelegt werden und es muss natürlich auch wieder für die richtige XML-Definition gesorgt werden. Danach folgt das Anlegen des Installationsskript `script.php` und es müssen SQL-Dateien für Installation, Deinstallation und Update erstellt werden. Dann folgen die MVC-Patterns im Backend. Schließlich benötigen wir noch ein Backend-Formular und es muss eine Table-Klasse angelegt werden. Zum Schluss werden noch Konfigurationsdialog, Helper-Klassen, Backend-Sprachdateien und Medien-Dateien benötigt. Im Frontend braucht man natürlich auch ein MVC-Pattern und Sprachdateien. Zuletzt wird das gesamte Projekt gezippt, damit man es über Joomla! installieren kann.⁵⁹

2.4.3.3. Fazit:

Dieser kurze Einblick in die Komponenten-Programmierung sollte nur zeigen, dass man mit Joomla! beliebig komplexe Programmieraufgaben versuchen kann. Es gibt natürlich schon viele Vorlagen, die man nur geringfügig adaptieren muss um zu scheinbar selbst erstellten guten Ergebnisse zu kommen, aber es obliegt dem Ehrgeiz der SchülerInnen, ob sie etwas wirklich Neues selbst versuchen wollen oder etwas schon Vorhandenes einfach für eigene Zwecke abändern wollen. Das Ergebnis wird in beiden Fällen für allfällige Kunden eindrucksvoll sein.

⁵⁸ Das komplette Projekt findet man bei David Jardin: Joomla! 2.5. Professionelle Webentwicklung. München: 2012. S. 323-364.

⁵⁹ Das komplette Listing findet man auf: <https://github.com/SniperSister/ComJobs>

2.5. Unterrichtspraktikum in Salzstiegl:

2.5.1. Vorbemerkung:

Ich hatte das Glück, dass ich während der Beschäftigung mit meiner Diplomarbeit im Mai 2013 ein Unterrichtspraktikum in Salzstiegl absolvieren durfte, wo ich genau die Inhalte meiner Diplomarbeit in der Praxis erproben konnte. Es arbeiteten zwei Kollegen mit mir im Team, wobei sich eine Befürchtung gleich bestätigte: Die Informatik-Kenntnisse der angehenden LehrerInnen sind teilweise erschreckend gering. So kannte ein Kollege überhaupt nicht den Begriff CSS und daher folgerichtig auch nicht den Zusammenhang mit HTML; auch von Datenbanken hatte er keine Ahnung. Er arbeitete dann schwerpunktmäßig nur im Klickbereich von Joomla! und konnte in keiner Weise in die Tiefe gehen.

2.5.2. Arbeitsplan:

Wir sollten jeden Tag andere SchülerInnen acht Stunden lang betreuen, um ihnen wesentliche Erkenntnisse über unser Stoffgebiet zu vermitteln. Ich konnte zum Glück eine SchülerInnengruppe zwei Tage betreuen, wodurch ich ein wenig erproben konnte, was eigentlich hängen bleibt, wenn man es einmal durchnimmt.

Für die 16 Stunden habe ich zunächst eine allgemeine Einführung in Joomla! vorgesehen und dann das Anlegen einer Datenbank und Arbeiten mit SQL. Ich habe das Hanke-Beispiel (siehe meine Arbeit ab Seite 33) probiert und es ist gut angekommen.

Vom Lehrer wurden wir darauf hingewiesen, dass wir nicht HTML und CSS machen sollen, da dies schon im Unterricht erledigt wurde.

Es hat sich in der Praxis gezeigt, dass die SchülerInnen von Joomla! begeistert sind, da man sehr rasch zu guten Ergebnissen kommt. Der Zusammenhang zwischen Installation und Datenbank hat aber bereits für eine kleine Überforderung gesorgt, auch das Arbeiten mit FTP war für die SchülerInnen neu. So habe ich die Installation selbst vorgenommen, wodurch schon ein wichtiger Lerneffekt verloren gegangen ist. Dafür muss man sich in der Praxis tatsächlich genug Zeit nehmen, da es keinen Sinn macht mit Joomla! zu arbeiten, wenn die SchülerInnen nicht einmal Joomla! installieren können.

Das Veröffentlichen von Beiträgen hat dann sehr gut geklappt, es hat sich aber wieder gezeigt, dass die SchülerInnen zwar Begriffe kennen, von diesen Begriffen aber keinen wirklichen Begriff haben: So kennen sie das Wort Struktur, es ist ihnen aber nicht klar, wie man komplexe Inhalte über HTML sinnvoll strukturieren kann. Sie kennen das Wort Layout, es ist ihnen aber nicht der Unterschied zwischen Layout und Formatierung klar. So habe ich schon alleine daraus für meine Unterrichtsarbeit gelernt, dass ich alle wichtigen Begriffe genau erklären werde, damit es zu keinen Missverständnissen kommen kann.

Obwohl die SchülerInnen einige Monate lang mit HTML und CSS gearbeitet haben, war ihnen der Zusammenhang dieser beiden Beschreibungssprachen immer noch nicht klar. HTML dient für Struktur und Inhalt, CSS für Layout und Formatierung. Es hat mich einige Zeit und viel Kraft gekostet, bis diese Grundsätze verinnerlicht waren.

Das nächste Problem ergab sich mit den Datenbanken. Ich habe jetzt ein gutes Gefühl dafür, wie lange man ca. brauchen wird, um die Idee von Datenbanken den SchülerInnen zu vermitteln. Ich Salzstiegl war die Zeit dafür jedenfalls viel zu kurz: Datenbanken bestehen aus Tabellen, Tabellen aus Datensätzen und diese aus Datenfeldern. Allein dieser Input ist nicht sehr nachhaltig in den SchülerInnenköpfen geblieben und ich musste auch diese Grundlagen einige Male wiederholen. Der Zusammenhang zwischen diesen Größen ist aber sehr wichtig, da die SchülerInnen wissen müssen, wann sie eine Tabelle anlegen müssen und wann ein Datenfeld. Die Übung von Hanke hat hier einige Klarheit geschaffen und auch zu einer gewissen Zufriedenheit der SchülerInnen. In meinem Konzept ist aber diese Übung nur ein Baustein zu einer vertieften Kenntnis der Datenbanken und diese war in der kurzen Zeit leider nicht zu erreichen.

Eine weitere Übung war die Erstellung eines Backups der Joomla!-Website. Auch das ist eine grundlegende Fertigkeit, die von Anfang an beherrscht werden muss, da es sehr ineffizient wäre, lange an einer Website zu arbeiten, von dieser nie ein Backup zu machen und am Ende des Tages aufgrund eines unvorhersehbaren Fehlers die gesamte Website zu zerstören. So wäre die ganze Arbeit umsonst gewesen. So ist also die Erstellung eines Backups eine der ersten Kompetenzen, die mit Joomla! beherrscht werden müssen. Auch hier hat sich gezeigt, dass die SchülerInnen trotz der

überschaubaren Schritte, die für ein Backup nötig sind, ziemlich überfordert waren. Auch dafür muss man wesentlich mehr Zeit einrechnen als mir zu Verfügung stand.

2.5.3. Fazit:

16 Stunden sind gar nicht so wenig, wenn man bedenkt, dass man bei einem Fach mit zwei Wochenstunden maximal 80 Stunden zur Verfügung hat, von denen aber garantiert einige entfallen, sodass man mit 16 Stunden vermutlich ein gutes Viertel der Unterrichtszeit eines Schuljahres zur Verfügung hat. Ich glaube, dass in geblockten acht Stunden am Tag weniger weitergeht als bei viermal zwei Stunden im Laufe von vier Wochen, da sich die SchülerInnen vielleicht doch auch zu Hause mit der Materie ein wenig beschäftigen oder zumindest Zeit haben, dass sich der Stoff etwas setzt und man nicht zu viel Neues auf einmal präsentiert bekommt. Mir ist nach diesem Unterrichtspraktikum auf alle Fälle klar geworden, dass man sich nicht zu viel Stoff vornehmen darf und dass man darauf achten muss, dass die zu erlernenden Kompetenzen sorgfältig aufgebaut und nach Möglichkeit miteinander verzahnt werden sollten. Man darf sich auch überhaupt nicht darauf verlassen, dass die SchülerInnen bestimmte Kompetenzen bereits KÖNNEN, sondern man muss regelmäßig kontrollieren, ob sie die von mir angedachte Kompetenzen tatsächlich beherrschen. In Salzstiegl musste ich erleben, dass die HTML- und CSS-Kompetenzen für einen sinnvollen Joomla!-Einsatz noch viel zu schwach waren, was dem Klassenlehrer offensichtlich gar nicht bewusst war.

2.6. Jahresplanung:

2.6.1. Vorbemerkung:

Gerade das Erlebnis Salzstiegl hat mir gezeigt, dass eine vernünftige Jahresplanung für einen erfolgreichen Unterricht unbedingt notwendig ist. Es ist für Informatik sinnvoll von 35 Doppelstunden von der 5.-7. Klasse pro Jahr auszugehen, in der 8. Klasse werden es dann nur ca. 30 Doppelstunden sein. Wenn ich also davon ausgehe, die SchülerInnen von der 5.-8. Klasse unterrichten zu können, stehen mir 135 Doppelstunden zur Verfügung, um das hier vorliegende Projekt umzusetzen. Ich werde in der Folge nur die Doppelstunden nennen und jeweils schreiben, was hier erledigt werden sollte.

2.6.2. Fünfte Klasse:

1-4: Einführung in Joomla!. Alle SchülerInnen sollen eine eigene Website haben, von dieser ein Backup mit Akeeba, beides auf einem externen Server über FTP erreichbar.

Die Website soll die Fächer der Klasse wiedergeben, in den jeweiligen Fächern sollen sinnvolle Beiträge gepostet werden. Einführung in GIMP, um eine leistungsfähige Bildbearbeitung kennenzulernen (in der Folge wird bei jeder Arbeit mit CSS auch mit GIMP gearbeitet, um diese Kompetenz am Laufen zu halten).

5-8: Arbeit mit HTML und CSS. Die SchülerInnen sollen einfache Websites mit HTML herstellen können und mit CSS für die Formatierung und das Layout sorgen können. Diese Website soll mit der Joomla!-Website verlinkt werden.

9-12: Arbeit mit PHP und SQL. Die SchülerInnen können Datenbanken mit Tabellen bestücken und diese über PHP mit Daten füttern. Sie arbeiten dabei mit phpMyAdmin, SQL und PHP. Natürlich müssen sie die Grundstruktur der Formulare mit HTML herstellen und diese mit CSS gestalten, was eine Wiederholung von 5-8 bedeutet. Datenbanken wurden auch schon bei der Joomla!-Einführung erstellt, deshalb ist dies eine Wiederholung eines Teiles von 1-4.

13-16: Vertiefung in Joomla!. Es werden bestehende Templates umgestaltet (Wiederholung von HTML, CSS und PHP), Arbeit mit dem Template Allrounder, bestehende Komponenten und Module in die Website eingebaut.

17-20: Einfache Programmieraufgaben mit PHP.

21-24: Vertiefung in die Datenbanken. SQL-Schulung mit MySQL und Access.

25-28: CSS-Gestaltung von Joomla! Arbeit mit der template.css

Die restlichen Stunden werden für Tests, Wiederholungen oder Vertiefungen benötigt.

2.6.3. Sechste Klasse:

In der sechsten Klasse würde ich den Stoff der fünften Klasse noch einmal wiederholen und gegebenenfalls vertiefen. Hier soll die Basis für neue Aufgaben gelegt werden. Man darf auch nicht vergessen, dass in der 6. Klasse in der Regel noch einige neue SchülerInnen dazukommen, die dadurch die Chance haben, in den laufenden Betrieb einzusteigen.

Nach meiner bisherigen Unterrichtserfahrung wird viel zu wenig Wert auf eine Festigung des Lehrstoffs gelegt, wodurch die Unterrichtsarbeit aber weitgehend sinnlos bleibt, da fast nichts hängenbleibt und keine erkennbaren Kompetenzen erreicht wurden.

2.6.4. Siebente Klasse:

1-4: Programmieren eigener Templates für Joomla!

5-8: Programmieren eigener Komponenten für Joomla!

9-13: Vertiefung in der HTML- und CSS-Programmierung.

14-18: Vertiefung in der PHP- und SQL-Programmierung.

19-23: Arbeiten mit JavaScript.

24-28: Erster Kontakt mit dem Joomla!-Framework

Die restlichen Stunden werden für Tests, Wiederholungen oder Vertiefungen benötigt.

2.6.5. Achte Klasse:

1-8: Arbeit mit dem Joomla!-Framework

9-12: Webshop-Systeme

13-18: Arbeit mit relationalen Datenbanken für die Organisation von Firmen.

19-22: Komplexe Programmieraufgaben mit beliebigen Sprachen

Die restlichen Stunden werden für Tests, Wiederholungen oder Vertiefungen benötigt.

2.6.6. Nachbemerkung:

Wie schon erwähnt, setze ich von der Unterstufe her Textverarbeitung, Präsentation und Arbeiten im Internet (E-Mail, Recherche, soziale Netzwerke usw.) voraus. Eine Tabellenkalkulation und ein CAD-Programm sollte im Mathematikunterricht oder in Geometrisch Zeichnen, bzw. Darstellender Geometrie eingesetzt werden.

2.7. Anforderungen für die LehrerInnen-Ausbildung

Ein großes Problem für mein Konzept stellt die derzeitige LehrerInnen-Ausbildung für Informatik dar. Hier ist für mich kein Zusammenhang zu erkennen, sondern meiner Meinung nach scheint Beliebigkeit an der Tagesordnung zu stehen. Die Beliebigkeit dürfte sich – so erschien es mir in meiner Ausbildung - nach den Vorlieben der

UniversitätslehrerInnen richten, die sich vielleicht auf kein effektives Konzept einigen wollen, das wirklich gut ausgebildete AHS-LehrerInnen hervorbringen könnte.

Man müsste die Ausbildung nach bestimmten Prämissen orientieren. Einerseits sollte man sich einigen, was AHS-LehrerInnen nun tatsächlich minimal beherrschen sollten. Das sollte sich nun wieder daran orientieren, was für SchülerInnen interessant zu lernen wäre. Und das müsste sich letztendlich einerseits daran orientieren, was die Wirtschaft und Gesellschaft von AbsolventInnen einer AHS in Informatik erwarten, und andererseits daran, was PädagogInnen glauben, dass SchülerInnen über diesen großen Bereich tatsächlich wissen sollten. Wie man sieht, steht hier eine gewaltige Koordinationaufgabe vor uns, wo sich LehrerInnen in Zusammenarbeit mit der Wirtschaft auf einen Lehrplan einigen müssten und UniversitätslehrerInnen dann die StudentInnen für die Vermittlung dieser Lehrinhalte, Kompetenzen und auch Einstellungen ausbilden müssten.

Von all dem ist aber seltsamer Weise nichts erkennbar. Das, was an der Universität unterrichtet wird, hängt nur sehr lose mit Teilen des Lehrplans zusammen. AHS-LehrerInnen halten sich häufig auch kaum an den Lehrplan und unterrichten einfach nach ihren persönlichen Vorlieben, was prinzipiell nicht schlecht ist, da sie in diesen Bereichen bestimmt die größte Kompetenz haben. Gerade in der AHS müsste man aber doch auch auf eine gewisse Breite des Angebot achten, da sonst der Bildungsauftrag einer Allgemeinbildung für die Informatik nicht annähernd erreicht werden würde. Manchmal ist es ihnen auch völlig egal, ob SchülerInnen tatsächlich ihre Kompetenzen am Computer steigern konnten. Es kommt sogar vor, dass manche InformatiklehrerInnen ihre SchülerInnen am Computer einfach „still beschäftigen“, die SchülerInnen vergnügen sich dann mit Computerspielen oder sozialen Medien. All diese Informationen habe ich nicht nur durch eigene Anschauung gewonnen, sondern zusätzlich auch von StudienkollegInnen mehrfach erfahren. Es gibt rund um den Informatikunterricht jedenfalls arge Missstände, die teilweise mit der nicht vorhandenen Ausbildung älterer LehrerInnen zusammenhängen, die mit wenigen Wochenendseminaren die Lehrberechtigung erlangt haben, und teilweise auch mit der mangelnden Motivation der Schulbehörde und der Universität, wirklich durchgreifende Verbesserungen zu erreichen, die den Anforderungen des 21. Jahrhunderts gerecht werden würden. Solche Kommentare mögen in einer wissenschaftlichen Arbeit

unpassend wirken, es geht aber in der Wissenschaft auch darum sich der Wahrheit anzunähern und dann darf man die Augen vor Missständen nicht verschließen, die freilich noch viel zu wenig offiziell dokumentiert wurden. Dass es daneben auch viele sehr engagierte InformatiklehrerInnen gibt, kann nicht zu oft betont werden.

3. Schluss

3.1. Abstract (Zusammenfassung)

Ich habe in dieser Arbeit ein Konzept für ein Schulbuch erstellt, nach dem man in der Oberstufe der AHS unterrichten könnte.

Ich gehe dabei von dem Content Management System Joomla aus, mit dem jeder Schüler eine Website erstellt, mit der er einen Überblick über seine Arbeit in den einzelnen Schulfächern geben muss. Textverarbeitung und Präsentationen müssen von der Unterstufe her schon beherrscht werden. Arbeiten aus diesen Bereichen sollen als PDF auf den eigenen Server gespeichert werden, um sie mit der Website zu verlinken. Mit einer Tabellenkalkulation muss in Mathematik gearbeitet werden, CAD-Programm könnten ebenfalls in Mathematik oder in Geometrisch Zeichnen bzw. Darstellender Geometrie verwendet werden.

Der Schwerpunkt des Informatikunterrichts liegt auf den Datenbanken (MySQL und Access), auf HTML, XML, CSS (plus Bildbearbeitung GIMP), PHP, SQL und JavaScript und daher auch auf Joomla!, da dieses CMS genau die genannten Bereiche abdeckt und benötigt.

Ich halte es für sinnvoll, die Anforderungen des Informatikunterrichtes nicht zu breit anzulegen, da sonst nichts bei den SchülerInnen als dauerhafte Kompetenz hängenbleibt. Man soll lieber einen schmalen Bereich mit bewussten Lücken (Geschichte der Informatik, Hardware, technische Programmierung, kaufmännische Informatik) abdecken, diesen aber durch laufende Wiederholungen so festigen (aus diesem Grund wird auch in der sechsten Klasse der gleiche Stoff wie in der fünften Klasse gemacht), damit die SchülerInnen mit einer echten Kompetenz im Webdesign- und Datenbank-Bereich die Schule verlassen.

3.2. Nachbemerkung

Am 10. Juni 2013 wurde interessanter Weise an Volksschulen folgende E-Mail verschickt:

Sehr geehrte Frau Direktor, sehr geehrter Herr Direktor!
Ich bitte Sie die Nachricht dieses Mal an das **gesamte Lehrerteam** weiter zu leiten.
Herzlichen Dank!
Liebe Kolleginnen, liebe Kollegen!

Ich bin für die Computerbetreuung und auch den Informatikunterricht in unserer Schule <http://www.marywardschulen.com/kremsphs/> seit meinem 1. Dienstjahr zuständig. Durch diese langjährige Tätigkeit und die ständige Diskussion zum Thema Schule habe ich die Seite www.informatiklehrer.at ins Leben gerufen. Daher auch mein INFO-Mail mit einer Umfrage: Es ist zur Zeit der **Informatikunterricht** in den Schulstufen 5-8 leider **nicht einheitlich** geregelt. Weiterführende Schulen und Lehrbetriebe können daher auf kein informationstechnisches **Grundwissen** zurückgreifen. Von **gut ausgebildet** bis **kaum Wissen vorhanden** ist sehr viel möglich. Auch weiterführende Schulen bieten Informatik nur teilweise an. So kann es vorkommen, dass **Maturanten** nie ein **Informatikgrundwissen** vermittelt bekommen. Aus diesem Grund bitte ich Euch die Umfrage zum Thema <http://www.informatiklehrer.at/index.php?id=43> zu unterstützen. Es würde mich freuen, wenn die Nachricht an Freunde, Bekannte und Interessierte weiterleitet wird. **Facebook** wäre auch ein gutes Mittel zur Verbreitung! [Kommetar: Warum verbreitet Herr Graf sein Anliegen nicht selbst über soziale Medien?] Wer mehr Interesse zum Thema mitbringt kann sich auch gerne auf der Seite registrieren, oder ist schon registriert. Für **Informatiker** gibt es dann **3 weitere wichtige Umfragen!** Vielleicht ist es möglich die gesammelten **Umfragedaten**, dann an die richtigen Stellen zu bringen und ein **Umdenken** zu erwirken. Ich bedanke mich im Voraus für die Unterstützung
Euer
Christian Graf

Die Website der Schule von Herrn Graf wurde mit Joomla 1.7 erstellt (aktuell ist derzeit 3.0) und die Website <http://www.informatiklehrer.at> für sein Projekt mit Typo3 4.5. Diese Website konnte Herr Graf allerdings nicht alleine erstellen, da er einen Experten als Webmaster zugezogen hat, der eine Firma für EDV-Lösungen anbietet:

<http://www.mknet.at> (auf dessen Website konnten übrigens im Juni 2013 die Umlaute nicht angezeigt werden). All das macht nicht besonders viel Mut, wenn man bedenkt, dass auch manche der engagiertesten InformatiklehrerInnen Österreichs offensichtlich über eine relativ geringe Kompetenz verfügen...

Die Lösung dieses Problems kann nur in einer reformierten Universitätsausbildung liegen und in regelmäßigen Kontrollen, wo darauf geachtet wird, wie die LehrerInnen eigentlich Informatik unterrichten.

Ich selbst habe beim Hospitieren erlebt, dass eine Lehrerin, die an ihrer Schule auch Kustodin ist, lediglich die SchülerInnen aufgefordert hat, sich Videos auf Youtube anzuschauen, um herauszufinden, wie bestimmte Programme funktionieren. Nach einigen Stunden habe ich sie gebeten, ob ich nicht einmal eine normale Stunde sehen könnte, weil ich dabei doch mehr lernen würde. Ihre Antwort: „Das sind meine normalen Stunden.“

3.3. Was ist gelungen, was ist misslungen?

Bei der Beschäftigung mit diesem Thema ist mir bewusst geworden, dass es sich eigentlich um ein sehr weites Feld handelt, wenn ich auch eine Eingrenzung des eigentlichen Informatikstoffes laut Lehrplan vorgenommen habe. Damit wurde noch deutlicher, wie absurd die Idee ist, den eigentlichen Stoff in einer sinnvollen Art und Weise abzuhandeln.

Die praktischen Übungen mit SchülerInnen haben mir gezeigt, dass es vielen Schwierigkeiten bereitet auch nur wenige Schritte fehlerlos durchzuführen (z. B. die Anmeldung an einer gerade erstellten Datenbank oder die Änderung eines Formulars und damit auch die damit verbundene Änderung einer Datenbank, die die Informationen der Formularfelder übernehmen soll). Diese Arbeiten müssen einige Male geübt werden, bis sie rasch und einigermaßen fehlerfrei durchgeführt werden können. Genau so muss man den SchülerInnen auch ausreichend Zeit geben, gemacht Fehler selbst zu finden und zu korrigieren. Diese Zeit hat man aber eigentlich nicht, wenn man den Stoff gut gefestigt an den Schüler bringen will. Dieses Dilemma wird mich wohl in der Praxis häufig beschäftigen; ich fürchte, dass es dafür auch keine zufriedenstellende Lösung gibt. Unterrichten ist wohl immer eine Gratwanderung zwischen Festigen und Vermittlung von neuem Lehrstoff.

Mein Scheitern ist insofern schon programmiert, da eine Änderung der Studienpläne auch wegen fehlender finanzieller Möglichkeiten derzeit nicht realistisch ist, genauso wie die Änderung des AHS-Lehrplans. Es wird auch unrealistisch bleiben zu denken, dass sich die Mehrzahl der LehrerInnen tatsächlich an Lehrpläne halten wird und nicht eher Informatik nach ihren eigenen Vorlieben unterrichten wird. Genauso unrealistisch ist die Idee, dass die Arbeit von LehrerInnen tatsächlich von Fachleuten evaluiert und – falls notwendig – kritisiert und verbessert wird.

Dennoch halte ich es für wichtig, offensichtliche Missstände aufzuzeigen und Verbesserungsvorschläge zu kommunizieren. Vielleicht gibt es ja einen Fortschritt in kleinen Verbesserungen, wenn sich immer mehr KollegInnen in einer konstruktiven Weise zusammenschließen, um den eigenen Unterricht zu optimieren.

3.4. Code-Beispiele

3.4.1. HTML:

Für das Grundgerüst in HTML schlage ich folgenden Code vor:

```
<!DOCTYPE html>
<html lang="de">
  <!--
    Bild einfügen: <img src="" width="" height="" alt="" title="" />
    Link einfügen: <a href="http://www.orf.at" rel="extern" title="Fernsehen">ORF</a>
    Anker einfügen: <a href="#ziel">Ziel</a> <h2 id="ziel">Ziel</h2>
  -->
  <head>
    <meta charset="UTF-8" />
    <title>
    </title>
    <link rel="stylesheet" href="css.css" />
  </head>
  <body>

  </body>
</html>
```

Wie man sieht, gibt es hier im Kommentar-Bereich des head Kommentarzeilen, die man einfach nur kopieren muss, damit die SchülerInnen ein Bild einfügen können, einen Hyperlink oder einen Anker. Dieser Code ist in HTML 5 abgefasst.

Zur Übung würde ich auch den folgenden Code zur Verfügung stellen, damit die SchülerInnen auch noch älteren HTML-Code kennen- und deuten lernen:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="de">
  <!--Bild einfügen: 
  Hyperlink: <a href="http://www.orf.at">ORF</a> -->
  <head>
    <title>CSS-Übung 1</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <!--Formmail Dateien müssen mit charset=ISO-8859-1 gespeichert werden.-->
    <link rel="stylesheet" type="text/css" media="screen" href="/">
  </head>
  <body>
    <!-- Beginn container -->
    <div id="container">
      <!-- Beginn kopf -->
      <div id="kopf">
        <h1>Kopf</h1>
        <p>Ein Beispieltext für den Kopfbereich.</p>
      </div>
      <!-- Ende kopf -->
      <!--Beginn links -->
      <div id="links">
        <h3>Links</h3>
        <p>Ein Beispieltext für die linke Spalte.</p>
      </div>
      <!-- Ende links -->
```

```

<!-- Beginn rechts -->
<div id="rechts">
<h3>Rechts</h3>
<p>Ein Beispieltext für die rechte Spalte.</p>
</div>
<!-- Ende rechts -->
<!-- Beginn inhalt -->
<div id="inhalt">
<h2>Der Inhalt</h2>
<p>Ein Beispieltext für den wichtigen Inhalt.</p>
</div>
<!-- Ende inhalt -->
<!-- Beginn impressum -->
<div id="impressum">
<h4>Impressum</h4>
<p>Ein Beispieltext für das Impressum.</p>
</div>
<!-- Ende impressum -->
</div>
<!-- Ende container -->
</body>
<!-- Ende sichtbarer Teil-->
</html>
</body>
</html>

```

Dieser Code soll dann von den SchülerInnen in HTML 5 umgeschrieben werden, wobei die entsprechenden DIV-Container in header, nav, article, footer und sidebar umbenannt werden müssen. Diese Übung ist nicht sehr schwierig, bringt aber schon ein wenig Gefühl für HTML.

3.4.2. CSS:

Als CSS-Code würde ich für den Beginn diesen Code zur Verfügung stellen:

```

/* Die Standardeinstellungen des Browsers werden zurückgesetzt */
* {
margin: 0px;
}
/* Hintergrundfarbe für den ganzen sichtbaren Bereich */
body {
background-color: olive;
}
/* Hintergrundfarben für die einzelnen DIVs */
#container {
background-color: purple;
/* Aussenabstand zum body */
margin: 20px;
}
#kopf {
background-color: lime;
}
#links {
background-color: yellow;
/* Lass es auf der linken Seite schweben! */
float: left;
/* Eine fixe Breite für #links */
width: 100px;
}
#rechts {
background-color: yellow;
/* Lass es auf der rechten Seite schweben! */
float: right;
/* Eine fixe Breite für #rechts */

```

```

width: 100px;
}
#inhalt {
background-color: fuchsia;
}
#impressum {
background-color: aqua;
clear: both;
}

```

Eine wichtige Arbeitsaufgabe ist nun die Erstellung eines schönen Farbensets. Da können die SchülerInnen auf der einen Seite nach ihrem persönlichen Geschmack ihr Glück versuchen, oder mit Hilfe entsprechender Websites eine gute Farbzusammenstellung finden. In diesem Bereich müssen sie einige Routine antrainiert bekommen, da die Farbzusammenstellung ein besonders wichtiger Punkt im Webdesign ist.

3.4.3. PHP, SQL und Datenbanken:

All diese Bereiche werden durch den folgenden Code abgedeckt. Zunächst geht es um ein Formular, in das Datensätze eingegeben werden können:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="de">
<head>
<title>Adressen eingeben</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<link rel="stylesheet" type="text/css" href="phpkid.css">
</head>
<body>
<p><a href="zeige_adressen.php">Adressen anzeigen</a></p>
<h3>Adressen eingeben</h3>
<form action="eingabe_adressen.php" method="post">
Vorname: <input type="text" name="Vorname"><br>
Name: <input type="text" name="Name"><br>
Str: <input type="text" name="Str"><br>
PLZ: <input type="text" name="PLZ"><br>

Ort: <input type="text" name="Ort"><br>
Tel: <input type="text" name="Tel"><br>
E-Mail: <input type="text" name="EMail"><br>
Webadresse: <input type="text" name="WWW"><br><br>
Notizen:<br>
<textarea cols="60" rows="4" name="Notizen">
</textarea><br>
<input type="submit" value="Daten eintragen" name="submit">
</form>
<?php
if (!empty($_POST["submit"])) {
$db = @mysqli_connect("localhost", "root", "") or die("Verbindung zu MySQL gescheitert");
mysqli_set_charset($db, 'utf8');
@mysqli_select_db($db, "team") or die("Datenbankzugriff gescheitert!");
$sql="INSERT INTO adressen " .
"(id, Vorname, Name, Str, PLZ, Ort, Tel, EMail, " .
"WWW, Notizen) VALUES (" , $_POST[Vorname], " .

```

```

    "$_POST[Name]', '$_POST[Str]', '$_POST[PLZ]', " .
    "$_POST[Ort]', '$_POST[Tel]', '$_POST[EMail]', " .
    "$_POST[WWW]', '$_POST[Notizen]')";
    if (mysqli_query($db, $sql)) {
        echo "<p>Abfrage erfolgreich!</p>";
    } else {
        echo "<p>Abfrage nicht erfolgreich!</p>";
    }
    mysqli_close($db);
}
?>
<p><a href="zeige_adressen.php">Adressen anzeigen</a></p>
</body>
</html>

```

Interessant dabei ist, dass die SchülerInnen auch eine Datenbank über phpMyAdmin anlegen müssen, in der die gleichen Tabellen und Datenfelder definiert werden. Das ist sehr leicht möglich, wenn man ihnen den SQL-Code zur Verfügung stellt:

```

CREATE TABLE adressen (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    Vorname VARCHAR(20),
    Name VARCHAR(20),
    Str VARCHAR(20),
    PLZ CHAR(5),
    Ort VARCHAR(30),
    Tel VARCHAR(25),
    EMail VARCHAR(30),
    WWW VARCHAR(30),
    Notizen TEXT
) DEFAULT CHARACTER SET utf8;

```

Nun könnte man aber verlangen, dass der Tabellename und die Datenfelder geändert werden. Dies wäre eine sehr praxisbezogene Übung, da man für einen Kunden auch nicht eine Datenbank neu programmieren würde, sondern mit einer ähnlichen Datenbank beginnen wird, um diese dann nach den Wünschen des Kunden umzugestalten.

Die Arbeit wird noch etwas schwieriger, wenn man die Daten zum Schluss in einer Tabelle ausgeben lässt. Dann muss diese Tabelle auch entsprechend angepasst werden:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="de">
<head>
<title>Adressen anzeigen</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<link rel="stylesheet" type="text/css" href="phpkid.css">
</head>
<body>
<p><a href="eingabe_adressen.php">Adressen eingeben</a></p>
<h3>Adressen anzeigen</h3>
<div>
<?php
$db = @mysqli_connect("localhost", "root", "") or die("Verbindung zu MySQL gescheitert!");
mysqli_set_charset($db, 'utf8');

```

```

@mysqli_select_db($db, "team") or die("Datenbankzugriff gescheitert!");
$sql = "SELECT * FROM adressen";
$result = mysqli_query($db, $sql);
$zeilen = mysqli_num_rows($result);
echo "<p>$zeilen Einträge:</p>\n";
// Tabelle definieren:
echo "<table border='1' cellpadding='0'>\n";
// Kopfzeile der Tabelle anlegen:
echo "<tr><th>id</th><th>Vorname</th>" .
"<th>Name</th><th>Str</th><th>PLZ</th>" .
"<th>Ort</th><th>Tel</th><th>EMail</th>" .
"<th>WWW</th><th>Notizen</th></tr>\n";
// while-Schleife Anfang
while ($row = mysqli_fetch_assoc($result)) {
    echo "<tr>"; // Zeile erzeugen
    // foreach Anfang:
    foreach ($row as $key => $value) {
        echo "<td>$value&nbsp;</td>\n";
    } // foreach Ende
    echo "</tr>"; // Zeile schließen
} // while Ende
echo "</table>\n"; // Tabelle schließen
mysqli_close($db);
?>
</div>
</body>
</html>

```

Natürlich könnte man diesen Code auch wieder auf HTML 5 umschreiben lassen und mit CSS ansprechend layoutieren und formatieren lassen.

Für einen Test oder eine Prüfung könnte man auch den Code erklären lassen.

3.4.4.Joomla!-Code:

Es würde den Rahmen dieser Arbeit sprengen, wenn wir jetzt für Joomla!-Projekte Code an dieser Stelle veröffentlichen würden. Der bisher veröffentlichte Code ermöglicht das Üben von Grundlagen. Danach kann man beliebigen Joomla!-Code beginnen zu analysieren und zu bearbeiten. Natürlich wird man dabei immer wieder auf Grenzen stoßen, die man aber bestimmt weggoogeln kann, da das Internet ja bekanntlich alle Grenzen aufhebt.

3.5. Wozu braucht man GIMP?

Noch ein letztes Wort zur Bildbearbeitung. Natürlich ist Phtoshop längst ein Synonym für Bildbearbeitung geworden und das Gratisprogramm GIMP kann diesem Profiprogramm nicht das Wasser reichen. Dennoch plädiere ich für den Einsatz von GIMP, weil die meisten Anforderungen für das Webdesign auch mit diesem Programm

zu erfüllen sind, weil es ähnlich wie Joomla durch Gratismodule aus dem Internet erweiterbar ist und – weil es gratis ist und die meisten Schulen ohnedies unter notorischem Geldmangel leiden.

GIMP ist relativ leicht zu erlernen⁶⁰ und tatsächlich sehr leistungsfähig. Natürlich kann man auch Bilder mit Picasa bearbeiten und veröffentlichen, komplexere Anwendungen müssen aber GIMP überlassen bleiben. Ideal wäre es, wenn auch der Lehrer für Bildnerische Erziehung zeitweise mit den SchülerInnen am Computer arbeiten würde, um sie ein wenig in GIMP zu schulen.

Jedenfalls müssen die Grafiken für das Internet optimiert werden, die SchülerInnen sollten Buttons entwerfen können und auch Farbverläufe sollten kein Problem sein. Ein anspruchsvolle Website hat auch meist grafische Elemente integriert, die ebenfalls mit einer Bildbearbeitung eigens hergestellt wurden.

All diese Bereiche sollten zusätzlich zum bereits beschriebenen Programm geübt werden. Hier muss man aufpassen, dass man sich nicht verzettelt, da GIMP tatsächlich das anspruchsloseste Werkzeug in unserem Zusammenhang bleiben muss, da die Bildbearbeitung mit den Grundaufgaben der Informatik schon relativ wenig zu tun hat (deshalb würde sie auch besser in den Unterricht für Bildnerische Erziehung passen). Das Gleiche gilt für Videoschnittprogramme. Das Einbinden von Videos bekommt im modernen Webdesign auch eine immer größere Bedeutung.

⁶⁰ Überhaupt mit der richtigen Literatur. Z. B.

Bettina K. Lechner: Gimp. Für digitale Fotografie, Webdesign und kreative Bildbearbeitung. München: 2009.
Michael Gradias: Gimp 2.8. Einstieg und Praxis. München: 2013.

3.6. Lebenslauf

Schulische Ausbildung:

1993 - 1997 Volksschule Bendagasse in 1230 Wien

1997 - 2001 Unterstufe im RGORG23 Anton Krieger Gasse in 1230 Wien

2001 - 2005 Oberstufe, informatik Zweig RGORG23 Anton Krieger Gasse

Juni 2005 Matura

Universitäre Ausbildung:

WS 2005 Beginn des Lehramtsstudium UF Informatik und Informatikmanagement UF Psychologie und Philosophie

WS 2007 Beginn des Lehramtsstudium UF Bewegung und Sport UF Informatik und Informatikmanagement

März 2013 Beginn der Diplomarbeit

3.7. Literaturverzeichnis

Boenigk, Cornelia und Friedhelm Betz: PHP 5. München 2004.

Caba, Helmut und Claudio Landerer: Neuer Informatik-Lehrplan. Was ist neu am neuen Informatik-Lehrplan 5. Klasse AHS? In: CD Austria. 3/2005. S. 11ff.

Dankwerth, Kerstin und Dana Boldt: Joomla! 2.5 Templates. Mit umfangreichem Praxisteil. Heidelberg: 2012.

Hagelkruys, Dominik: Diplomarbeit „Barrierefreiheit als Thematik im österreichischen Informatikunterricht. Wien: 2012.

Hanke, Johann-Christian: PHP und MySQL für Kids. Heidelberg: 2010.

Hilger, Gertraud, Maria Kiener: klar text Deutsch5. Wien: 2010. . S. 115f.(Schulbuch Nr. 140586)

Jardin, David: Joomla! 2.5. Professionelle Webentwicklung. München: 2012.

Koch, Daniel: Das große Buch zu Joomla! 2.5. Vom schnellen Einstieg zum professionellen Webauftritt. Düsseldorf: 2012.

Lawson, Bruce und Remy Sharp: HTML 5. München: 2011.

Lechner, Bettina K.: Gimp. Für digitale Fotografie, Wedesign und kreative Bildbearbeitung. München: 2009.

Maurice, Florence: Mobile Webseiten. Strategien, Techniken, Dos und Don'ts für Webentwickler. München: 2012.

Gradias, Michael: Gimp 2.8. Einstieg und Praxis. München: 2013.

Prevezanos, Christoph: Joomla 3. Websites erstellen und gestalten. München: 2013.

Radtke, Angie: Joomla! Templates entwickeln. Barrierefreie & attraktive Designs von Konzept bis Umsetzung. München: 2011.

Reibold, Holger: phpMyAdmin kompakt. Saarbrücken: 2009.

Schroer, Michael: Web Content Management mit PHP und MySQL. Bonn: 2004.

Schürmann, Tim: Praxiswissen Joomla! 2.5.- Köln: 2012.

Sigos, David: CSS für Kids. Heidelberg: 2012.

Weyand Birgit, Monika Justus, Michael Schratz (Hrsg.): Auf unsere Lehrerinnen und Lehrer kommt es an. Geeignete Lehrer/-innen gewinnen, (aus-)bilden und fördern. Essen: 2012.

Wieken, John-Harry: erfolgreich SQL lernen. München: 2012.

Websites:

<http://www.ahs-informatik.com/>
<http://www.akeebabackup.com/>
<http://www.apachefriends.org/de/xampp.html>
<http://api.joomla.org>
<http://www.arge-informatik.tsn.at/>
http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf
http://www.bmukk.gv.at/medienpool/22837/reifepruefung_ahs_mrp.pdf
<http://www.bomots.de>
<https://www.box.com/s/mrkqjercce0rp93hdx>
<http://bplaced.net>
<http://www.brain-media.de>
<http://colorschemedesigner.com/>
<http://www.csszengarden.com/tr/deutsch/>
<http://de.selfhtml.org/html/formulare/>
<http://eass.bplaced.net/45-Joomla!>
<http://www.ecdl.at>
<http://extensions.joomla.org>
<http://www.filezilla.de/download.htm>
<https://github.com/SniperSister/ComJobs>
<http://www.gym1.at/schulinformatik/buecher/ahs-informatik.pdf>
www.informatiklehrer.at
<http://it1018.egger.ac/>
<http://www.joomlaos.de>
<http://www.lernvid.com/downloads/allrounder-Joomla!-16-template.html>
<http://www.marywardschulen.com/kremsphs/>
<http://www.mknet.at>
<http://topstyle-lite.soft-ware.net/download.asp>
<http://square7.ch>
<https://support.google.com/picasa/?hl=de>
<http://www.youtube.com>